IN THE UNITED STATES PATENT AND TRADEMARK OFFICE ON APPEAL FROM THE EXAMINER TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	Jack C. H. Chung et al.
Serial No.:	10/055,098
Filing Date:	January 22, 2002
Group Art Unit No.	3623
Examiner:	Andre D. Boyce
Confirmation No.	8102
Title:	INTEGRATED DECISION SUPPORT FRAMEWORK FOR COLLABORATIVE PRODUCT DEVELOPMENT

MAIL STOP APPEAL BRIEF - PATENT
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

CORRECTED APPEAL BRIEF

Appellants have appealed to the Board of Patent Appeals and Interferences ("Board") from the decision of the Examiner mailed December 18, 2006, finally rejecting Claims 1-22. Appellants filed a Notice of Appeal and Pre-Appeal Brief Request for Review on March 8, 2007, with authorization to charge the statutory fee of $500.00. Appellants file this Corrected Appeal Brief in response to the Notification of Non-compliance that was mailed June 6, 2007. This Corrected Appeal Brief is also filed in response to a Panel Decision dated March 28, 2007, which indicates that there is at least one issue for Appeal.

Real Party In Interest

This Application is currently owned by Unified Graphics Systems Corporation (UGS Corp.) as indicated by:

an assignment recorded on 01/22/2002 from inventors Jack C.H. Chung, Jia-Yi (NMI) Wang, and Chien-Tai (NMI) Wu to Electronic Data Systems Corporation, in the Assignment Records of the PTO at Reel 012529, Frame 0741 (6 pages);

an assignment recorded on 06/28/2004 from Electronic Data Systems Corporation to Unified Graphics Systems PLM Solutions Incorporated, in the Assignment Records of the PTO at Reel 015354, Frame 0900 (6 pages);

an assignment recorded on 06/28/2004 from Unified Graphics Systems PLM Solutions Incorporated to Unified Graphics Systems Corporation, in the Assignment Records of the PTO at Reel 014782, Frame 0440 (10 pages); and

an assignment of security interest recorded on 06/28/2004 from Unified Graphics Systems Corporation to JPMorgan Chase Bank, as Administrative Agent, in the Assignment Records of the PTO at Reel 014782, Frame 0450 (9 pages).

Related Appeals and Interferences

To the knowledge of Appellants' counsel, there are no known interferences or judicial proceedings that will directly affect or be directly affected by or have a bearing on the Board's decision regarding this Appeal.

Status of Claims

Claims 1-22 are pending in this Application with Claims 1-22 being rejected pursuant to a Final Office Action mailed December 18, 2006 (the "Final Office Action"). All pending claims are shown in the Claims Appendix, attached hereto, along with an indication of the status of those claims.

Claims 1-3, 5, 9, 11-14, 16, 20, and 22 are rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 6,102,958 issued to Meystel et al ("*Meystel*"). Claims 4, 6-8, 10, 15, 17-19 and 21 are rejected under 35 U.S.C. §103(a) as being unpatentable over *Meystel*, in view of U.S. Patent No. 6,826,541 issued to Johnston et al ("*Johnston*"). For the reasons discussed below, Appellants respectfully submit that these rejections are improper and should be reversed by the Board. Accordingly, Appellants presents Claims 1-22 for Appeal.

All pending claims are shown in Appendix A, attached hereto, along with an indication of the status of those claims. A copy of *Meystel* is attached as Evidence Appendix 1 and a copy of *Johnston* is attached as Evidence Appendix 2.

Status of Amendments

All amendments submitted by Appellants have been entered by the Examiner.

Summary of Claimed Subject Matter

Essentially, in accordance with one example embodiment of the present invention, an integrated decision support framework is provided, whereby different types of decision-drivers from numerous sources can be converted into a unified decision network including, for example, both mathematical and node-edge graph representations. A graph-theoretic algorithm may be applied to the large problem (unified decision network) to detect and separate strongly-connected components. The strongly-connected components represent sub-problems that must be solved simultaneously. A dependency propagation technique may be used to properly order the sub-problems so they can be processed and solved sequentially and correctly. Each strongly-connected component (small sub-problem) can be delegated to a suitable decision generator depending on the types of relations included in the component. For example, a numerical solution algorithm may be used to solve the ordered, numerical relations sub-problems, an algebraic solution algorithm may be used to solve the ordered, geometric relations sub-problems, and a logical inference engine (algorithm) may be used to solve the ordered, logical relations sub-problems. Solutions thus derived can be propagated to the next stage of the decision resolution process until a complete problem is solved. (*Specification*, page 8, lines 6-30).

FIGURE 2 illustrates an example method 100 for integrated decision support that may be used to implement one example embodiment of the present invention. For example, method 100 may be a computer-executed software application used to implement an example integrated decision support framework application such as 16a or 16b in FIGURE 1. At step 102, all diverse inputs (e.g., decision-drivers) for a product development process are received. For example, at step 102, some or all of (but not necessarily limited to) the following diverse types of input information may be received with respect to a collaborative product development process: option selections; linear relations; dependencies; production rules; logical relations; inequality expressions; geometric constraints; etc. At step 104, the received inputs can be converted into a unified decision network that can include both node-edge graph representations and mathematical representations of the received input information. By compiling these conversions and representations into a unified decision network (e.g., a

unified set of inputs), all of the received inputs may be taken into account before an overall decision is derived. (*Specification*, page 10, line 20 through page 11, line 9).

FIGURE 3 illustrates how certain of the received decision inputs can be converted into node-edge graph representations.

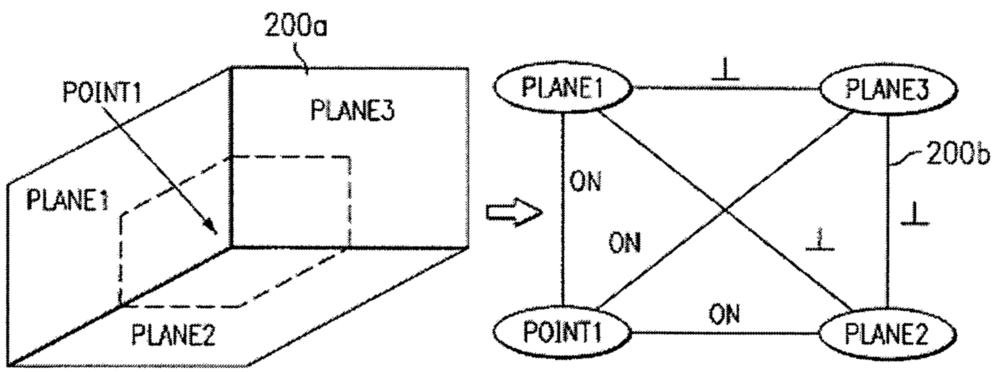


FIG. 3

As shown, the three-dimensional structure 200a can be represented as a node-edge graph 200b. The geometric entities of structure 200a may be represented as nodes in graph 200b, and the constraints imposed on structure 200a may be represented as edges in graph 200b. For example, the geometric entities Point 1, Plane 1, Plane 2 and Plane 3 in structure 200a are represented as the nodes in graph 200b. The edges in graph 200b describe the constraints or limitations imposed on the structure 200a, or the relationships between the nodes. As illustrated by graph 200b, Point 1 is constrained to be located on Plane 1, Plane 2 and Plane 3, while Plane 1, Plane 2 and Plane 3 are constrained to be perpendicular to each other. (*Specification*, page 11, lines 10-25).

At step 104 (as mentioned earlier), certain of the received input decision-drivers can also be expressed as mathematical expressions (e.g., conditional equations, inequality equations, equality equations, difference equations, etc.). For example, a received input including an equality relation between the size of a particular pin and the size of the hole needed to accept the pin may be expressed mathematically as "Hole_Size = Pin_Size + 5mm". Also, for example, a received input including an inequality relation as to the cost of a

part may be expressed mathematically as "Total_Cost < 1000 dollars". A received input including a conditional relation for a part may be expressed mathematically, for example, as: "if Loading_Capacity > X, use Roller_Bearing, else use Ball_Bearing". In other words, the geometric entities for such received inputs may be represented as variables, and the constraints and dimensions imposed may be represented as mathematical equations. (*Specification*, page 11, line 26 through page 12, line 12).

At step 106, after the received inputs are converted into a unified decision network (e.g., decision inputs now expressed as one large problem including graph representations and mathematical representations), the large problem can be decomposed into a group of smaller sub-problems. For example, a graph-theoretic algorithm may be applied to the one large problem in order to decompose it into a group of smaller sub-problems. For the mathematical representations of the large problem, a graph-theoretic algorithm (e.g., See Serrano, D., and Gossard, D. C., "Combining Mathematical Models With Geometric Models In CAE Systems," Proc. ASME Computer in Eng. Conf., Chicago, IL, July 1986.) may be applied to match each mathematical equation with a unique variable to prevent over-constraining. After the equation-variable matching is completed, the dependency and coupling between all equations can be derived, which leads to the breakup of the total equation set into smaller groups. The smaller groups of equations can then be solved sequentially and efficiently using suitable computer-implemented solution techniques, for example, the Newton-Raphson computation method or the Modified Gram-Schmidt computation method. (*Specification*, page 12, line 13 through page 13, line 3).

In order to decompose the graph representations of the large problem into a group of smaller sub-problems, the structure of the node-edge graph may be analyzed. An analysis of the graph's structure shows (e.g., see graph 200b) that certain of the nodes may be considered terminal nodes, because all of the edges incident upon them can be used to drive those nodes. In other words, the graph may be decomposed and thus simplified by removing terminal nodes from the graph. As such, a node and its connecting edges may be removed from a graph, if the total number of degrees of freedom of the edges is less than or equal to that of the node. For example, in graph 200b, node Point 1 is incident on three planes, which generally provides a stable solution. Consequently, for decomposition purposes, node Point 1

and all three incident edges can be removed from graph 200b. FIGURE 4 illustrates how graph 200b can be decomposed into a set of algebraically solvable components, by removing a set of terminal nodes (e.g., Point 1, Plane 1, Plane 3, Plane 2) from graph 200b. (*Specification*, page 13, line 14 through page 14, line 2).

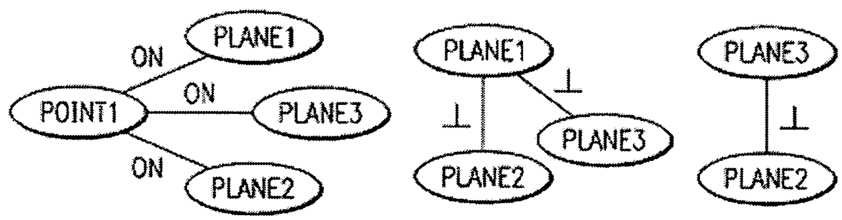


FIG. 4

At step 108, as part of the large, unified problem decomposition process, certain algorithms (e.g., graph-matching algorithms, Tarjan's algorithm, tri-connectivity algorithms, etc.) may be applied to the algebraically solvable sub-problems in order to detect the strongly-connected components (e.g., coupling between the different equations). Notably, tri-connected graphs include exactly three paths between any pair of distinct vertices. Consequently, tri-connectivity algorithms may be suitable to use for ordering graph sub-problems, because most problems encountered in industry are not highly coupled (e.g., 2 or 3 couples at the most). (*Specification*, page 14, lines 3-14).

At step 110, once the decomposition process is completed and the set of smaller sub-problems has been identified, a dependency propagation technique may be used to place the sub-problems in a proper order so that they can be sequentially and correctly processed and solved. In general, dependency propagation is a relational technique that records the relationships between the dimensions or parameters of the sub-problems, and then maintains these relationships as the dimensions or parameters are changed. For example, using dependency propagation to order the algebraically solvable components (FIG. 4) of the graph 200b shown in FIGURE 3, a solution sequence for graph 200b can be (maintaining the relationships or dependencies between components): Plane 2 -> Plane 3; Plane 3 -> Plane 1; and Plane 1 -> Point 1. As a result of the decomposition process, an ordered list of small problems (e.g., 2 or 3 relations) is thus ready to be solved. Notably, for an overall decision in

a practical industry environment, there may be thousands of these groups of small problems (e.g., 2 or 3 simultaneous relations per group) that need to be solved. (*Specification*, page 14, line 15 through page 15, line 4).

At step 112, each group of small problems can be solved sequentially and a group at a time. For example, a numerical solution algorithm (e.g., implemented in software) may be used to solve the purely numerical relations problems, an algebraic solution algorithm may be used to solve the purely geometric relations problems, and a logical inference algorithm may be used to solve the purely logical relations problems sequentially in accordance with the listed order. (*Specification*, page 15, lines 5-13).

At step 114, once an attempt has been made to solve the entire, ordered list of sub-problems, any sub-problem that may not be solved (e.g., a conflict exists), if any, may be identified. At step 116, if any such sub-problem is identified, the input or decision-driver related to that sub-problem may be re-specified, and the present method can return to step 104 to re-process that re-defined input. Otherwise, at step 120, an overall decision may be reported as complete. The integrated decision support method 100 may be re-iterated with new or re-specified inputs until a complete problem is deemed solved. (*Specification*, page 15, lines 14-25).

With regard to the independent claims currently under Appeal, Appellants provide the following concise explanation of the subject matter recited in the claim elements. For brevity, Appellants do not necessarily identify every portion of the Specification and drawings relevant to the recited claim elements. Additionally, this explanation should not be used to limit Appellants' claims but is intended to assist the Board in considering the Appeal of this Application.

For example, independent Claim 1 recites the following:

> A method for integrated decision support (e.g., Figure 2; *Specification,* page 10, line 20 through page 15, line 25), comprising the steps of:
>
> receiving a plurality of decision inputs (e.g., Figure 2, step 102; *Specification,* page 10, line 26 through page 11, line 2);
>
> converting a first plurality of said received decision inputs to a plurality of graph representations (e.g., Figure 2, step 104; Figure 3; *Specification,* page 11, lines 2-25);
>
> converting a second plurality of said received decision inputs to a plurality of mathematical representations (e.g., Figure 2, step 104; *Specification,* page 11, line 26 through page 12, line 12);
>
> decomposing said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems (e.g., Figure 2, step 106; Figure 4; *Specification,* page 12, line 13 through page 14, line 2);
>
> detecting a plurality of strongly-connected components associated with said plurality of sub-problems, each of said plurality of strongly-connected components representing a connection between at least two of said plurality of sub-problems (e.g., Figure 2, step 108; *Specification,* page 14, lines 3-14); and
>
> solving said plurality of sub-problems (e .g., Figure 2, step 112; *Specification,* page 15, lines 5-13).

As another example, independent Claim 11 recites the following:

> Software for integrated decision support (e.g., Figure 1, reference numeral 16a; *Specification,* page 8, line 31 through page 10, line 19), the software being embodied in a computer-readable medium and when executed operable to:
>
> receive a plurality of decision inputs (e.g., Figure 2, step 102; *Specification,* page 10, line 26 through page 11, line 2);
>
> convert a first plurality of said received decision inputs to a plurality of graph representations (e.g., Figure 2, step 104; Figure 3; *Specification,* page 11, lines 2-25);
>
> convert a second plurality of said received decision inputs to a plurality of mathematical representations (e.g., Figure 2, step 104; *Specification,* page 11, line 26 through page 12, line 12);
>
> decompose said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems (e.g., Figure 2, step 106; Figure 4; *Specification,* page 12, line 13 through page 14, line 2);
>
> detect a plurality of strongly-connected components associated with said plurality of sub-problems, each of said plurality of strongly-

connected components representing a connection between at least two of said plurality of sub-problems (e.g., Figure 2, step 108; *Specification,* page 14, lines 3-14); and

solve said plurality of sub-problems (e.g., Figure 2, step 112; *Specification,* page 15, lines 5-13).

As another example, independent Claim 12 recites the following:

A computer-implemented system for integrated decision support (e.g., Figure 1, reference numeral 10; *Specification,* page 8, line 31 through page 10, line 19), comprising:

a processor (e.g., Figure 1, reference numeral 14; *Specification,* page 8, line 31 through page 10, line 19); and

a data storage device coupled to said processor (e.g., Figure 1, reference numeral 20; *Specification,* page 8, line 31 through page 10, line 19), said processor operable to:

receive a plurality of decision inputs (e.g., Figure 2, step 102; *Specification,* page 10, line 26 through page 11, line 2);

convert a first plurality of said received decision inputs to a plurality of graph representations (e.g., Figure 2, step 104; Figure 3; *Specification,* page 11, lines 2-25);

convert a second plurality of said received decision inputs to a plurality of mathematical representations (e.g., Figure 2, step 104; *Specification,* page 11, line 26 through page 12, line 12);

decompose said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems (e.g., Figure 2, step 106; Figure 4; *Specification,* page 12, line 13 through page 14, line 2);

detect a plurality of strongly-connected components associated with said plurality of sub-problems, each of said plurality of strongly-connected components representing a connection between at least two of said plurality of sub-problems (e.g., Figure 2, step 108; *Specification,* page 14, lines 3-14); and

solve said plurality of sub-problems (e.g., Figure 2, step 112; *Specification,* page 15, lines 5-13).

As another example, independent Claim 22 recites the following:

A system for integrated decision support (e.g., Figure 1, reference numeral 10; *Specification,* page 8, line 31 through page 10, line 19), comprising:

means for receiving a plurality of decision inputs (e.g., Figure 1, reference numeral 10; Figure 2, step 102; *Specification,* page 8, line 31 through page 10, line 19; *Specification,* page 10, line 26 through page 11, line 2);

means for converting a first plurality of said received decision inputs to a plurality of graph representations (e.g., Figure 1, reference numeral 10; Figure 2, step 104; Figure 3; *Specification,* page 8, line 31 through page 10, line 19; *Specification,* page 11, lines 2-25);

means for converting a second plurality of said received decision inputs to a plurality of mathematical representations (e.g., Figure 1, reference numeral 10; Figure 2, step 104; Figure 3; *Specification,* page 8, line 31 through page 10, line 19; *Specification,* page 11, line 26 through page 12, line 12);

means for decomposing said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems (e.g., Figure 1, reference numeral 10; Figure 2, step 106; Figure 4; *Specification,* page 8, line 31 through page 10, line 19; *Specification,* page 12, line 13 through page 14, line 2);

means for detecting a plurality of strongly-connected components associated with said plurality of sub-problems, each of said plurality of strongly-connected components representing a connection between at least two of said plurality of sub-problems (e.g., Figure 1, reference numeral 10; Figure 2, step 108; *Specification,* page 8, line 31 through page 10, line 19; *Specification,* page 14, lines 3-14); and

means for solving said plurality of sub-problems (e.g., Figure 1, reference numeral 10; Figure 2, step 112; *Specification,* page 8, line 31 through page 10, line 19; *Specification,* page 15, lines 5-13).

In this Appeal Brief, Appellants have argued certain dependent claims separately. With regard to these dependent claims, Appellants provide the following concise explanation of the subject matter recited in the claim elements. For brevity, Appellants do not necessarily identify every portion of the Specification and drawings relevant to the recited claim elements. Additionally, this explanation should not be used to limit Appellants' claims but is intended to assist the Board in considering the Appeal of this Application.

For example, dependent Claim 2 recites the following:

performing dependency propagation for said plurality of sub-problems (e.g., Figure 2; Page 8, lines 16-19; Page 14, lines 15 through Page 15, line 4).

Dependent Claim 13 recites certain similar features and operations.

As another example, dependent Claim 3 recites the following:

executing a graph-theoretic algorithm for a plurality of mathematical equations associated with said plurality of strongly-connected components to prevent over-constraining (e.g., Figure 2; Page 8, lines 12-14; Page 12, lines 13-27).

Dependent Claim 14 recites certain similar features and operations.

As another example, dependent Claim 4 recites the following:

> decomposing said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of mathematical equations and algebraically solvable graph components (e.g., Figure 4; Page 13, line 30 through Page 14, line 2; Page 14, line 26 through Page 15, line 4).

Dependent Claim 15 recites certain similar features and operations.

As another example, dependent Claim 5 recites the following:

> detecting a plurality of dependency relations within said plurality of sub-problems (e.g., Figure 2; Page 12, line 27 through Page 13, line 3; Page 14, line 15 through Page 15, line 4).

Dependent Claim 16 recites certain similar features and operations.

As another example, dependent Claim 6 recites the following:

> identifying a plurality of simultaneous equations within said plurality of sub-problems (e.g., Page 4, lines 17-21; Page 8, lines 12-16; Page 14, line 24 through Page 15, line 4).

Dependent Claim 17 recites certain similar features and operations.

As another example, dependent Claim 7 recites the following:

> solving a plurality of numerical sub-problems and a plurality of algebraic sub-problems (e.g., Figure 4; Page 8, lines 19-30; Page 13, line 30 through Page 14, line 2; Page 14, line 26 through Page 15, line 4).

Dependent Claim 18 recites certain similar features and operations.

As another example, dependent Claim 8 recites the following:

> solving a plurality of numerical relations sub-problems with a numerical solution algorithm; (e.g., Figure 2; Page 8, lines 19-30; Page 14, lines 5-13)
>
> solving a plurality of geometric relations sub-problems with an algebraic solution algorithm (e.g., Figure 2; Page 8, lines 19-30; Page 14, lines 5-13); and
>
> solving a plurality of logical relations sub-problems with a logical inference solution algorithm (e.g., Figure 2; Page 8, lines 19-30; Page 14, lines 5-13).

Dependent Claim 19 recites certain similar features and operations.

As another example, dependent Claim 10 recites the following:

> solving a plurality of simultaneous equations with a Newton-Raphson algorithm or Modified Gram-Schmidt algorithm (e.g., Figure 2; Page 12, line 27 through Page 13, line 3).

Dependent Claim 21 recites certain similar features and operations.

Grounds of Rejection to be Reviewed on Appeal

Are Claims 1-3, 5, 9, 11-14, 16, 20, and 22 anticipated under 35 U.S.C. § 102(b) by U.S. Patent No. 6,102,958 issued to Meystel et al. ("*Meystel*")?

Are Claims 4, 6-8, 10, 15, 17-19, and 21 unpatentable under 35 U.S.C. § 103(a) over *Meystel* in view of U.S. Patent No. 6,826,541 issued to Johnston et al. ("*Johnston*")?

Arguments

Claims 1-3, 5, 9, 11-14, 16, 20, and 22 are rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 6,102,958 issued to Meystel et al ("*Meystel*"). Claims 4, 6-8, 10, 15, 17-19 and 21 are rejected under 35 U.S.C. §103(a) as being unpatentable over *Meystel*, in view of U.S. Patent No. 6,826,541 issued to Johnston et al ("*Johnston*"). Copies of *Meystel* and *Johnston* are attached in the Evidence Appendix. For the reasons discussed below, Appellants respectfully submit that these rejections are improper and should be reversed by the Board.

I. The Rejection of Appellants' Claims under 35 U.S.C. § 102 is Improper

A. Standard

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987); M.P.E.P. § 2131. In addition, "[t]he identical invention <u>must</u> be shown in as complete detail as is contained in the . . . claims" and "[t]he elements <u>must</u> be arranged as required by the claim." *Richardson v. Suzuki Motor Co.*, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989); *In re Bond*, 15 U.S.P.Q.2d 1566 (Fed. Cir. 1990); M.P.E.P. § 2131 (*emphasis added*).

B. The *Meystel* Reference

Meystel relates to "a process control system that determines optimal trajectories (input controls) using multiresolutional analysis of acquired data. In contrast to conventional control systems, the present control system does not use a predetermined mathematical model or algorithm to define the process in terms of a plurality of variables. Rather, the present system acquires system data and stores the data in a multiresolutional data structure. A knowledge base is created which can be searched at varying levels of resolution for determining optimal process trajectories. The continual addition of data to the data structure allows for continual top-down refinement of the determined trajectories and bottom-up improvement and updating of the system representation." (*Meystel*, Abstract).

"The present invention comprises a multiresolutional decision support system (MDSS) for plant performance enhancement. MDSS implements selected features of multiresolutional control theory to assist in the process of making optimal, or cost minimizing, decisions for both process control and design. MDSS is able to integrate information from diverse sources or subsystems and, without using models or equations, to formulate plans or rules of system operation which provide the best performance according to a set of criteria of interest, subject to the constraints of available information. One of the main features of MDSS is an ability to generalize precise information (acquired data) into successively more abstract levels of resolution, which allows a significant reduction in the complexity of analysis of alternatives. Such analysis can, therefore, be performed via successive refinement of increasingly accurate projections within a more constrained envelope at each level of resolution. Once the multiresolutional structure of consecutively generalized data is constructed, a top-down "navigation" is possible which constitutes the core of the MDSS functioning." (*Meystel*, column 8, lines 29-50).

"MDSS efficiently analyzes distributed system information obtained as a result of on-line data acquisition. Acquired data is organized in a hierarchy of scale and scope, permitting examination in a narrower envelope at each succeeding level of higher resolution. Each plan determined at a higher resolution is more detailed than at preceding (lower resolution) levels and is based on a more localized and precise examination of selected information. The hierarchical organization dramatically reduces the complexity of information storage and analysis. MDSS allows the use of a distributed modeling technique with information acquired and embedded into the model, to synthesize optimal strategies of operation." (*Meystel*, column 8, lines 51-63).

"Generally, the system 100 comprises a preprocessing subsystem 102, a representation subsystem 104, a planning or search subsystem 106, an arbitration subsystem 108, a prediction or optimization subsystem 110, and a stabilizing feedback subsystem 112. The plant or process 14 being controlled is monitored by a plurality of sensors, as previously discussed, which sensors provide data 16 for use by the system 100. The plant or process 14 includes several levels of data acquisition for a general multicomponent system running several concurrently actuated loops. The levels may be derived from physical considerations,

such as dependence of one loop on a satisfactory operation of another loop. The operation of the control system 100 is discussed below." (*Meystel*, column 15, lines 11-24).

"The plant or process 14 performance can be measured by a variety of yardsticks, including financial savings, the meeting of regulatory standards, maximization of the operational life of a plant, and minimization of the need for operator intervention (reliability). As long as the cost measures, however objective, are tangible and may be evaluated from measurements, they can be incorporated into the present control system 100, either separately or in a weighted combination. As opposed to prior art analytical approaches, there is no constraint on the simultaneous evaluation of cost by a variety of measures." (*Meystel*, column 15, lines 25-35).

"The overall strategy of optimization and successive refinement employed in MDSS is contingent upon the creation of a multiresolutional representation. The key requirements of the system 100 are that it must contain a generalized representation of system capabilities at a resolution (level) that allows global optimization without violating computational restraints and that each level must contain a more dense (precise) representation than the previous level. If this is achieved, then search at each successive level of resolution generates a more precise solution of the optimization problem. If prior experiences are informationally rich, the process of optimization is based on such rich experiences, and if prior experiences are poor, interpolation and extrapolation may be required." (*Meystel*, column 15, lines 36-50).

"A multiresolutional representation may be visualized as a hierarchical store of information, incorporating appropriately quantized descriptions of achievable system responses. The information is organized by precision into levels of increasingly detailed description. Algorithms for the incorporation of new information into the hierarchy are considered part of the system's knowledge. Thus, a finite set of information about the system 100 is assembled, which, in association with search and retrieval algorithms, answers the same questions as the more familiar mathematical abstraction or model, viz, what is the effect on measurable or identifiable plant variables of a particular excitation. Learning is implemented by collecting and integrating experiences and by subsequent random testing in the intervals between experiences. Information about the same objects is represented at

several appropriate levels of resolution (nested, hierarchical, local decomposition is a complicated but apt description of this principle). The ability to analyze gross behavior, as well as local perturbations is also built in for efficient analysis of alternatives. The source of a model can be tested or carried out on any of an existing plant, an exiting dynamic model of more or less arbitrary complexity, or a custom-built model with elements that include rules, look-up tables, equations, and empirical data. Subjective and inconsistent, as well as traditional cost or penalty criteria are locally evaluated and incorporated into a model for future analysis. The overall strategy is thus one which places the burden of determining and achieving performance goals upon the control system, as opposed to a system designer. The present invention reinforces the feedback control component of a control system, using the byproduct of the planning subsystem 106 and the planning component (using learning algorithms and the feedback derived from system operation), the control system 100 is provided with the attributes of intelligence." (*Meystel*, column 15, line 51 through column 16, line 17).

"The major operational steps of the present invention are preprocessing and data analysis, multiresolutional model design, cost evaluation and process improvement. The first step, preprocessing and data analysis, provides an adequate description of the plant or process 14 by constructing and validating a distributed multiresolutional provisional model of the plant or process 14. The multiresolutional model may be validated using a multi-valued graph representation, as described in more detail below. Cost evaluation may be performed using a sliding window to compute the costs of operation of the process in a tangible form. The sliding window permits the application of the model for process improvement which can then be measured in a consistent manner. The width of the sliding window is determined by the actual discretization of the available information and the required accuracy of results. The process improvement step represents the continual improvement of the process by applying search and optimization algorithms to the distributed model. Critical but previously unknown relationships of the process may be discovered during the preprocessing of information for the model." (*Meystel*, column 16, lines 18-39).

C. Claims 1, 9, 11, 12, 20, and 22 are patentable over *Meystel*

Independent Claim 1 of the present application recites:

> A method for integrated decision support, comprising the steps of:
> receiving a plurality of decision inputs;
> converting a first plurality of said received decision inputs to a plurality of graph representations;
> converting a second plurality of said received decision inputs to a plurality of mathematical representations;
> decomposing said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems;
> detecting a plurality of strongly-connected components associated with said plurality of sub-problems, each of said plurality of strongly-connected components representing a connection between at least two of said plurality of sub-problems; and
> solving said plurality of sub-problems.

Appellants respectfully submit that *Meystel* does not disclose, either expressly or inherently, each and every element of Appellants' Claim 1.

For example, *Meystel* does not disclose, teach, or suggest "converting a second plurality of said received decision inputs to a plurality of mathematical representations," as recited in Claim 1. Rather, *Meystel* is concerned with regulating plant performance using a multi-valued graph representation, without generating mathematical representations. In the Final Office Action, the Examiner identifies Column 10, lines 9-21 of *Meystel* as disclosing Appellants' step of "converting a second plurality of said received decision inputs to a plurality of mathematical representations." Relevant portions of *Meystel* disclose that "[a] first organizational subsystem 20 acquires all inputs based information, such as the data 16, and organized such information for use by a modeling subsystem 22." (Column 9, lines 36-39). "The modeling subsystem 22 determines functional relationships of the organized data, which are searched by a behavior generation subsystem 24." (Column 9, lines 39-41). Thus, *Meystel* merely discloses the organization of data within the data structure for improved searching. The portion of the reference cited by the Examiner merely clarifies that an "associative fusion of data module 32

forms a statement of rules from the data stored and organized." (Column 10, lines 8-10). "The rules differ in the degree of belief which is assigned based on the repetitiveness of the particular experiences (statistical) or trustworthiness of the particular source of information (expert)." (Column 10, lines 10-13). "The associative fusion of data module 32 comprises means for extraction of goal-independent data from the stored goal-dependent "experiences" in the form of generalized rules, or components of an automata transition function and state-output functions." (Column 10, lines 13-18). Thus, the cited portion of *Meystel* merely discloses the extraction of data from the data structure based on generalized statistical or source-based rules. Even if these statistical or source-based rules are used to extract data from the data structure of *Meystel*, there is no disclosure in *Meystel* of "converting a second plurality of said received decision inputs to a plurality of mathematical representations," as recited in Claim 1.

In fact, several portions of *Meystel* specifically state that mathematical representations are not used. For example, in the Abstract, *Meystel* discloses that "the system does not use a predetermined mathematical model or algorithm." As another example, *Meystel* states, "In contrast to prior art control systems, the present invention does not use a predetermined mathematical model or algorithm which defines the process in terms of a plurality of variables." (Column 3, lines 2-5). "Rather, the present invention acquires system data and stores the data in a multiresolutional data structure." (Column 3, lines 5-6). In the Final Office Action, the Examiner counters:

> *Meystel et al* does not state that mathematical representations are not used. Rather *Meystel et al* simply states that it does not use a *predetermined* mathematical model or algorithm, but instead a fluid provisional data structure (column 3, lines 3-12) and that it *eliminates the stage* of mathematical abstraction and parameter identification (column 21, lines 59-63). As such, neither cited portion precludes *Meystel et al* from teaching Applicant's invention.

(Final Office Action, page 8). Appellants disagree. *Meystel* actually discloses that "[t]he present invention eliminates the stage of mathematical abstraction and parameter identification and instead, uses collected measurements (the data 16) to form a distributed multiresolutional knowledge representation which operated upon by search algorithms." (Column 21, lines 59-63). The use of "search and retrieval algorithms" to extract data from a data set, as identified

by the Examiner, is not analogous to "converting a second plurality of said received decision inputs to a plurality of mathematical representations," as recited in Claim 1. This fact is further emphasized by the touted advantage within *Meystel* that the multiresolutional decision support system "is able to integrate information from diverse sources or subsystems and, **with out using models or equations**, to formulate plans or rules of system operation which provide the best performance according to a set of criteria of interest, subject to the constraints of available information." (Column 8, lines 35-40, emphasis added). Accordingly, *Meystel* does not disclose, teach, or suggest "converting a second plurality of said received decision inputs to a plurality of mathematical representations," as recited in Claim 1.

As another example of the deficiencies of *Meystel*, Appellants submit that *Meystel* does not disclose, teach, or suggest "decomposing said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems," as recited in Claim 1. The Final Office Action asserts Column 9, lines 37-40 of *Meystel* discloses the recited claim elements. (Final Office Action, page 3). Specifically, the Final Office Action alleges that "the organizational subsystem 20 organizes information for use by modeling system 22." (Final Office Action, page 3). Appellants agree that the cited portion of *Meystel* states that "[a] first organizational subsystem 20 acquires all input based information, such as the data 16, and organizes such information for use by a modeling subsystem." (Column 9, lines 37-40). However, organizing information is not analogous to Appellants' step of decomposing **converted inputs** to **sub-problems**. In fact, conspicuously absent from the Final Office Action's allegation is any mention of sub-problems. Appellants submit that this omission is not coincidental since, as described above, *Meystel* does not use sub-problems or mathematical representations. At best, *Meystel* is a decision processor that simply integrates data into a data structure for searching. There is no disclosure in *Meystel* of "decomposing said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems," as recited in Claim 1.

For at least these reasons, Appellants respectfully submit that the rejection of independent Claim 1 and its dependent claims (including Claim 9) is improper and should be reversed by the Board.

The Examiner also relies on *Meystel* to reject independent Claims 11, 12, and 22. Appellants respectfully submit that *Meystel* does not disclose, teach, or suggest each and every element of Appellants' independent Claims 11, 12, and 22. For example, Claim 11 recites "software being embodied in a computer-readable medium and when executed operable to . . . convert a second plurality of said received decision inputs to a plurality of mathematical representations." Claim 11 also recites the software operable to "decompose said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems." As another example, Claim 12 recites a processor operable to "convert a second plurality of said received decision inputs to a plurality of mathematical representations." Claim 12 also recites that the processor is operable to "decompose said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems." Claim 22 recites "means for converting a second plurality of said received decision inputs to a plurality of mathematical representations." Claim 22 also recites "means for decomposing said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems." Thus, for reasons analogous to those discussed above with regard to Claim 1, Appellants respectfully submit that *Meystel* does not disclose, teach, or suggest each and every element set forth in Appellants' independent Claims 11, 12, and 22.

For at least these reasons, Appellants respectfully submit that the rejections of independent Claims 11, 12, and 22 and their respective dependent claims (including Claim 20 that depends on Claim 12) are improper and should be reversed by the Board.

D. Claims 2 and 13 are patentable over *Meystel*

Claims 2 and 13 depend upon independent Claims 1 and 12, respectively, which Appellants have shown above to be allowable. Accordingly, Claims 2 and 13 are not anticipated by *Meystel* at least because of this dependency.

Additionally, dependent Claims 2 and 13 recite elements that further distinguish the art. As an example, Claim 2 recites "performing dependency propagation for said plurality of sub-problems." Claim 13 recites certain analogous features and operations. In the Final Office Action, the Examiner identifies Column 10, lines 40-46 of *Meystel* as disclosing Appellants' recited claim elements. The cited portion merely states, however:

> The resolutional structure development module 34 comprises means fro developing a multiresolutional data (knowledge) structure based upon the "associative clusters", which can transform the provisional relational model into a multilevel hierarchical relational structure with any required number of levels of resolution, including embedded local dynamics of the plant or process 14.

(*Meystel*, column 10, lines 40-46). Although in quotations, the term "associative clusters" is not defined by *Meystel*. As guidance, however, *Meystel* provides:

> The associative fusion of data module 32 forms a statement of rules from the data stored and organized by the experience organizer module 30. The rules differ in the degree of belief which is assigned based upon repetitiveness of the particular experiences (statistical) or trustworthiness of the particular source of information (expert). The associative fusion of data module 32 comprises means for extraction of goal-independent data from the stored goal-dependent "experiences", in the form of **generalized rules, or components of an automata transition function and state-output functions**. The extraction means may comprise any standard algorithm and/or procedure for inverting experimental causalities into the statements of rules. Preferably, more sophisticated procedures are applied to the organized data, linked with algorithms and procedures of fusing the experiences into "associative clusters". Such clusters form a function oriented, goal-independent provisional model of the plant or process 14.

(Column 10, lines 8-24, emphasis added). Thus, *Meystel* merely discloses the organization of data within a data structure based on generalized rules or components of an automated transition function and state-output functions. *Meystel* does not disclose, teach, or suggest the step of "performing **dependency propagation** for said plurality of sub-problems," as recited in Claims 2 and 13.

For at least these reasons, Appellants respectfully submit that the rejections of dependent Claims 2 and 13 are improper and should be reversed by the Board.

E. Claims 3 and 14 are patentable over *Meystel*

Claims 3 and 14 depend upon independent Claims 1 and 12, respectively, which Appellants have shown above to be allowable. Accordingly, Claims 3 and 14 are not anticipated by *Meystel* at least because of this dependency.

Additionally, dependent Claims 3 and 14 recite elements that further distinguish the art. As an example, Claim 3 recites "executing a graph-theoretic algorithm for a plurality of mathematical equations associated with said plurality of strongly-connected components to prevent over-constraining." Claim 14 recites certain analogous features and operations. In the Final Office Action, the Examiner identifies Column 16, lines 42-45 of *Meystel* as disclosing Appellants' recited claim elements. The cited portion merely states, however:

> The preprocessing subsystem 102 is provided for preprocessing raw plant information or data 16 using hierarchical clustering algorithms in order to synthesize a multiresolutional knowledge base.

(*Meystel*, column 16, lines 42-45). Although discussing "hierarchical clustering algorithms," *Meystel* provides no indication that such hierarchical clustering algorithms are "graph-theoretic algorithms of mathematical equations," as recited in Claims 3 and 14.

Furthermore, Appellants have shown above that *Meystel* specifically states that mathematical representations are not used. For example, in the Abstract, *Meystel* discloses that "the system does not use a predetermined mathematical model or algorithm." As another example, *Meystel* states, "In contrast to prior art control systems, the present invention does not use a predetermined mathematical model or algorithm which defines the process in terms of a plurality of variables." (Column 3, lines 2-5). "Rather, the present invention acquires system data and stores the data in a multiresolutional data structure." (Column 3, lines 5-6). Touting an advantage of the multiresolutional decision support system disclosed in *Meystel*, *Meystel* stipulates that the system "is able to integrate information from diverse sources or subsystems and, **with out using models or equations**, to formulate plans or rules of system operation which provide the best performance according to a set of criteria of interest, subject to the

constraints of available information." (Column 8, lines 35-40, emphasis added). As such, *Meystel* does not disclose, teach, or suggest "a plurality of mathematical equations" and certainly does not disclose, teach, or suggest "executing a graph-theoretic algorithm for a plurality of mathematical equations associated with said plurality of strongly-connected components to prevent over-constraining," as recited in Claims 1.

For at least these reasons, Appellants respectfully submit that the rejections of dependent Claims 3 and 14 are improper and should be reversed by the Board.

F. Claims 5 and 16 are patentable over *Meystel*

Claims 5 and 16 depend upon independent Claims 1 and 12, respectively, which Appellants have shown above to be allowable. Accordingly, Claims 5 and 16 ar e not anticipated by *Meystel* at least because of this dependency.

Additionally, dependent Claims 5 and 16 recite elements that further distinguish the art. As an example, Claim 5 recites "detecting a plurality of dependency relations within said plurality of sub-problems." Claim 16 recites certain analogous features and operations. In the Final Office Action, the Examiner identifies Column 9, lines 40-42 of *Meystel* as disclosing Appellants' recited claim elements. The cited portion merely states, however:

> The modeling subsystem 22 determines functional relationships of the organized data, which are searched by a behavior generation subsystem 24.

(*Meystel*, column 9, lines 40-42). Although discussing "functional relationships," *Meystel* provides no indication that such functional relationships are analogous to Appellants' recited "dependency relations." Furthermore, Appellants have shown above that *Meystel* does not disclose, teach, or suggest "sub-problems" or that the data in the data structure is organized into "sub-problems." Again, Appellants submit that this omission is not coincidental since, as described above, *Meystel* does not use sub-problems or mathematical representations. At best, *Meystel* is a decision processor that simply integrates data into a data structure for searching.

There is no disclosure in *Meystel* of "detecting a plurality of dependency relations within said plurality of sub-problems," as recited in Claims 5 and 16.

For at least these reasons, Appellants respectfully submit that the rejections of dependent Claims 5 and 16 are improper and should be reversed by the Board.

II. The Rejections of Appellant's Claims under 35 U.S.C. § 103 are Improper

A. Standard

The Examiner rejects Claims 4, 6-8, 10, 15, 17-19 and 21 are rejected under 35 U.S.C. §103(a), which states:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The question raised under 35 U.S.C. § 103 is whether the prior art taken as a whole would suggest the claimed invention taken as a whole to one of ordinary skill in the art at the time of the invention. *See* 35 U.S.C. § 103(a). Accordingly, even if all elements of a claim are disclosed in various prior art references, which is certainly not the case here as discussed below, the claimed invention taken as a whole cannot be said to be obvious without some reason given in the prior art why one of ordinary skill in the art at the time of the invention would have been prompted to modify the teachings of a reference or combine the teachings of multiple references to arrive at the claimed invention.

The M.P.E.P. sets forth the strict legal standard for establishing a *prima facie* case of obviousness based on modification or combination of prior art references. "To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally

available to one of ordinary skill in the art, to modify the reference or combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references where combined) must teach or suggest all the claim limitations." M.P.E.P. § 2142, 2143. The teaching, suggestion or motivation for the modification or combination and the reasonable expectation of success must both be found in the prior art and cannot be based on an Appellant's disclosure. *See Id.* (citations omitted). "Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either explicitly or implicitly in the references themselves or in the knowledge generally available to one of ordinary skill in the art" at the time of the invention. M.P.E.P. § 2143.01. Even the fact that references *can* be modified or combined does not render the resultant modification or combination obvious unless the prior art teaches or suggests the desirability of the modification or combination. *See Id.* (citations omitted). Moreover, "To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. All words in a claim must be considered in judging the patentability of that claim against the prior art." M.P.E.P. § 2143.03 (citations omitted).

The governing Federal Circuit case law makes this strict legal standard even more clear.[1] According to the Federal Circuit, "a showing of a suggestion, teaching, or motivation to combine or modify prior art references is an essential component of an obviousness holding." *In re Sang-Su Lee*, 277 F.3d 1338, 1343, 61 U.S.P.Q.2d 1430, 1433 (Fed. Cir. 2002) (quoting *Brown & Williamson Tobacco Corp. v. Philip Morris Inc.*, 229 F.3d 1120, 1124-25, 56 U.S.P.Q.2d 1456, 1459 (Fed. Cir. 2000)). "Evidence of a suggestion, teaching, or motivation . . . may flow from the prior art references themselves, the knowledge of one of ordinary skill in the art, or, in some cases, the nature of the problem to be solved." *In re Dembiczak*, 175 F.3d 994, 999, 50 U.S.P.Q.2d 1614, 1617 (Fed. Cir. 1999). However, the "range of sources available . . . does not diminish the requirement for actual evidence." *Id.* Although a prior art device "may be capable of being modified to run the way the apparatus is claimed, there must be a suggestion or motivation in the reference to do so." *In re Mills*, 916 F.2d at 682, 16 U.S.P.Q.2d at 1432. *See also In re Rouffet,* 149 F.3d 1350, 1357, 47

[1] Note M.P.E.P. 2145 X.C. ("The Federal Circuit has produced a number of decisions overturning obviousness rejections due to a lack of suggestion in the prior art of the desirability of combining references.").

U.S.P.Q.2d 1453, 1457-58 (Fed. Cir. 1998) (holding a *prima facie* case of obviousness not made where the combination of the references taught every element of the claimed invention but did not provide a motivation to combine); *In Re Jones*, 958 F.2d 347, 351, 21 U.S.P.Q.2d 1941, 1944 (Fed. Cir. 1992) ("Conspicuously missing from this record is any evidence, other than the PTO's speculation (if that can be called evidence) that one of ordinary skill in the herbicidal art would have been motivated to make the modification of the prior art salts necessary to arrive at" the claimed invention.). Even a determination that it would have been obvious to one of ordinary skill in the art at the time of the invention to try the proposed modification or combination is not sufficient to establish a *prima facie* case of obviousness. *See In re Fine*, 837 F.2d 1071, 1075, 5 U.S.P.Q.2d 1596, 1599 (Fed. Cir. 1988).

In addition, the M.P.E.P. and the Federal Circuit repeatedly warn against using an Appellants' disclosure as a blueprint to reconstruct the claimed invention. For example, the M.P.E.P. states, "The tendency to resort to 'hindsight' based upon applicant's disclosure is often difficult to avoid due to the very nature of the examination process. However, impermissible hindsight must be avoided and the legal conclusion must be reached on the basis of the facts gleaned from the prior art." M.P.E.P. § 2142. The governing Federal Circuit cases are equally clear. "A critical step in analyzing the patentability of claims pursuant to [35 U.S.C. § 103] is casting the mind back to the time of invention, to consider the thinking of one of ordinary skill in the art, guided only by the prior art references and the then-accepted wisdom in the field. . . . Close adherence to this methodology is especially important in cases where the very ease with which the invention can be understood may prompt one 'to fall victim to the insidious effect of a hindsight syndrome wherein that which only the invention taught is used against its teacher.'" *In re Kotzab,* 217 F.3d 1365, 1369, 55 U.S.P.Q.2d 1313, 1316 (Fed. Cir. 2000) (citations omitted). In *In re Kotzab*, the Federal Circuit noted that to prevent the use of hindsight based on the invention to defeat patentability of the invention, the court requires the Examiner to show a motivation to combine the references that create the case of obviousness. *See id. See also, e.g., Grain Processing Corp. v. American Maize-Products,* 840 F.2d 902, 907, 5 U.S.P.Q.2d 1788, 1792 (Fed. Cir. 1988). Similarly, in *In re Dembiczak*, the Federal Circuit reversed a finding of obviousness by the Board, explaining that the required evidence of such a teaching,

suggestion, or motivation is essential to avoid impermissible hindsight reconstruction of an applicant's invention:

> Our case law makes clear that the best defense against the subtle but powerful attraction of hind-sight obviousness analysis is *rigorous application of the requirement for a showing of the teaching or motivation to combine prior art references*. Combining prior art references without evidence of such a suggestion, teaching, or motivation simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability—the essence of hindsight.

175 F.3d at 999, 50 U.S.P.Q.2d at 1617 (emphasis added) (citations omitted).

B. The *Johnston* Reference

Johnston relates to methods, systems, and computer program products for "facilitating user choices among complex alternatives [utilizing] conjoint analysis to simplify choices to be made by the user. A selector tool presents a user with a first and second series of choices relating to attributes of products or services available to the user. A utilities calculation engine calculates the relative utility of each of the products or services to the user and presents output to the user, which indicates the relative utility of each of the products or services. The user can then select the product or service that has the highest utility value for the user based on the calculated relative utility values." (*Johnston,* Abstract).

C. Claims 4, 6-8, 15, and 17-19

1. Claims 4, 6-8, 15, and 17-19 are Allowable over the Proposed *Meystel-Johnston* Combination

First, Appellants submit that the proposed *Meystel-Johnston* combination fails to teach, suggest, or disclose the combination of elements recited in Appellants' claims.

Claims 4 and 6-8 depend upon independent Claim 1, which Appellants have shown above to be allowable. Claims 15 and 17-19 depend upon independent Claim 12, which Appellants have shown above to be allowable. Accordingly, Claims 4, 6-8, 15, and 17-19

are not obvious over the proposed *Meystel-Johnston* combination at least because of their dependencies.

Additionally, dependent Claims 4, 6-8, 15, and 17-19 recite elements that further distinguish the art. For example, the proposed *Meystel-Johnston* combination does not disclose, teach, or suggest the following claim elements:

- "decomposing said converted first plurality of said received decision inputs and said second plurality of said received decision inputs to a plurality of mathematical equations and algebraically solvable graph components," as recited in Claim 4 (and similarly recited in Claim 15;
- "identifying a plurality of simultaneous equations within said plurality of sub-problems," as recited in Claim 6 (and similarly recited in Claim 17;
- "solving a plurality of numerical sub-problems and a plurality of algebraic sub-problems," as recited in Claim 7 (and similarly recited in Claim 18); and
- "solving a plurality of geometric relations sub-problems with an algebraic solution algorithm," as recited in Claim 8 (and similarly recited in Claim 19).

With respect to each of these claims the Examiner acknowledges in the Final Office Action that *Meystel* fails to disclose the recited claim elements. (*Final Office Action*, pages 5-7). Instead, as a basis for rejecting each of the above claim elements, the Examiner makes the following statement:

> Johnston et al discloses a method for facilitating choices among complex alternatives, wherein a statistical algorithm implemented involves calculation of regression coefficient (column 12, lines 38-42), including matrix analysis (column 14).

(*Final Office Action*, pages 5-7). The Examiner does not identify, however, how the disclosure of a statistical algorithm involving regression coefficients and matrix analysis is the same as performing the diverse operations recited in Appellants' Claims 4, 6-8, 15, and 17-19.

In fact, *Johnston* merely relates to "[a] utilities calculation engine calculates the relative utility of each of the products or services to the user [available to the user] and presents output

to the user, which indicates the relative utility of each of the products or services." (*Johnston*, Abstract). The portion of *Johnston* cited by the Examiner discusses a "statistical algorithm implemented by utilities calculation engine 202" that involves the "calculation of "regression" coefficients for inserting into an equation representing the quantitative values chosen by a individual to measure his or her judgment of the importance of the difference between some best value of each important "attribute" and some worst value of the "attribute" . . ." (*Johnston*, Column 12, lines 38-42). With regard to matrix analysis, *Johnston* merely discloses that the matrix analysis renders "final set of data to be analyzed." (Column 16, line 1). Specifically, "each row (of both the Y vector and the X matrix) represents a single response the "user" has given, the actual response being recorded in the Y vector." (Column 16, lines 23-25). "From this type data set, the B regression coefficients are calculated . . . Each B coefficient is paired with and thus modifies one "attribute." (Column 16, lines 33-41). Thus, *Johnston* merely discloses using matrix analysis to calculate utilities for attributes identified by a user.

However, there is no disclosure in *Johnston* of "algebraically solvable graph components" or of "decomposing . . . decision inputs to a plurality of mathematical equations and algebraically solvable graph components," as recited in Appellants' Claims 4 and 15. Additionally, there is no disclosure in *Johnston* that the calculation of regression coefficients through matrix analysis is analogous to "identifying a plurality of simultaneous equations within said plurality of sub-problems," as recited in Appellants' claims 6 and 17, "solving a plurality of numerical sub-problems and a plurality of algebraic sub-problems," as recited in Claims 7 and 18, or "solving a plurality of geometric relations sub-problems with an algebraic solution algorithm," as recited in Claims 8 and 19. These claim elements are absent from the disclosures of *Meystel* and *Johnston*.

For at least these reasons, Appellants respectfully submit that the rejections of dependent Claims 4, 6-8, 15, and 17-19 are improper and should be reversed by the Board.

2. One of Ordinary Skill in the Art Would not have been Motivated to Make the Proposed *Meystel-Johnston* Combination

Additionally, Appellants submit that the Examiner has not demonstrated the requisite teaching, suggestion, or motivation in *Meystel*, *Johnston*, or the knowledge generally available to those of ordinary skill in the art at the time of the invention to modify or combine *Meystel* and *Johnston* in the manner the Examiner proposes. The rejections are improper and should be reversed for at least this additional reason.

As discussed above in Section II.A of this Appeal Brief, the question raised under 35 U.S.C. § 103 is whether the prior art taken as a whole would suggest the claimed invention taken as a whole to one of ordinary skill in the art at the time of the invention. Accordingly, even if all elements of a claim are disclosed in various prior art references, which is certainly not the case here as discussed above, the claimed invention taken as a whole cannot be said to be obvious without some reason given in the prior art why one of ordinary skill at the time of the invention would have been prompted to modify the teachings of a reference or combine the teachings of multiple references to arrive at the claimed invention. In the Final Office Action, the Examiner relies upon *Meystel* as the primary reference and upon Johnston as the secondary reference. (*Final Office Action*, pages 5-7). Appellants respectfully submit, however, that the Examiner has merely pieced together disjointed portions of references, with the benefit of hindsight using Appellants' claims as a blueprint, in an attempt to reconstruct Appellants' claims.

According to the Examiner, "[b]oth *Meystel* and *Johnston* are concerned with improving decision making." (*Final Office Action,* pages 5-7). It is the Examiner's contention that "it would have been obvious to one of ordinary skill in the art at the time the invention was made" to modify *Meystel* "as seen in *Johnston*, as tool for making difficult decisions less complex (see *Johnston*, column 2, lines 21-26), thus making *Meystel* more effective and robust." (*Final Office Action,* pages 5-7). It appears that the Examiner has merely proposed alleged advantages of combining *Meystel* with *Johnston* (advantages which Appellants do not admit could even be achieved by combining these references in the manner the Examiner proposes). The Examiner has not pointed to any portions of the cited

references, however, that would teach, suggest, or motivate one of ordinary skill in the art at the time of invention to incorporate the system "for facilitating user choices among complex alternatives [utilizing] conjoint analysis to simplify choices to be made by the user" disclosed in *Johnston* with the multiresolutional decision support system disclosed in *Meystel*. In other words, the Examiner has not provided an explanation as to: (1) why it would have been obvious to one of ordinary skill in the art at the time of Appellants' invention (***without using Appellants' claims as a guide***) to modify the particular techniques disclosed in *Meystel* with the cited disclosure in *Johnston*; (2) how one of ordinary skill in the art at the time of Appellants' invention would have actually done so; and (3) how doing so would purportedly meet the limitations of Appellants' claims.

The mere possibility that a modification might improve *Meystel*, as the Examiner asserts, by making *Meystel* more "effective and robust" does not even remotely provide the required teaching, suggestion, or motivation to modify the teachings of *Meystel* Indeed, if it were sufficient for Examiners to merely point to a purported advantage of one reference and conclude that it would have been obvious to combine of modify that reference with other references simply based on that advantage (which, as should be evident from the case law discussed above, it certainly is not), then virtually any two or more references would be combinable just based on the fact the one reference states an advantage of its system. Of course, as the Federal Circuit has made clear and as discussed above, that is not the law.

Furthermore, Appellants respectfully submit that one of ordinary skill in the art would not have been motivated to make the proposed *Meystel-Johnston* combination. Even if both *Meystel* and *Johnston* relate to "improved decision making" as posited by the Examiner, *Meystel* specifically teaches away from the use of mathematical representations. For example, *Meystel* actually discloses that "[t]he present invention eliminates the stage of mathematical abstraction and parameter identification and instead, uses collected measurements (the data 16) to form a distributed multiresolutional knowledge representation which operated upon by search algorithms." (Column 21, lines 59-63). As a provided advantage, *Meystel* stipulates that the multiresolutional decision support system "is able to integrate information from diverse sources or subsystems and, **with out using models or equations**, to formulate plans or rules of system operation which provide the best

performance according to a set of criteria of interest, subject to the constraints of available information." (Column 8, lines 35-40, emphasis added). Accordingly, it would not have been obvious to one of ordinary skill in the art at the time of Appellants' invention to modify the multiresolutional decision support system disclosed in *Meystel* to include the calculation of regression coefficients and matrix analysis disclosed in *Johnston*.

For at least these reasons, Appellants respectfully submit that the proposed *Meystel-Johnston* combination is improper with respect to Appellants' Claims 4, 6-8, 15, and 17-19. Accordingly, the rejection of Appellants' claims over the proposed *Meystel-Johnston* combination should be reversed by the Board.

D. Claims 10 and 21 are Allowable over the Proposed *Meystel-Johnston* Combination

For the reasons discussed below, Appellants submit that the proposed *Meystel-Johnston* combination fails to teach, suggest, or disclose the combination of elements recited in Appellants' claims.

Claims 10 and 21 depend upon independent Claims 1 and 12, respectively, which Appellants have shown above to be allowable. Accordingly, Claims 10 and 21 are not obvious over the proposed *Meystel-Johnston* combination at least because of their respective dependencies.

Additionally, dependent Claims 10 and 21 recite elements that further distinguish the art. As an example, Claim 10 recites "solving a plurality of simultaneous equations with a Newton-Raphson algorithm or Modified Gram-Schmidt algorithm." Claim 21 recites certain analogous features and operations. In the Final Office Action, the Examiner acknowledges that *Meystel* fails to disclose the recited claim elements. However, the Examiner fails to identify any other reference that discloses the recited claim elements and that could be properly combined with *Meystel*. Instead, the Examiner states, "the Newton-Raphson algorithm and

Modified Gram-Schmidt algorithms are old and well known in the art, therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to include Newton-Raphson algorithm or Modified Gram-Schmidt algorithm in *Meystel*, as an efficient means of solving simultaneous equations, thus making *Meystel* more efficient and robust." (*Final Office Action*, page 7).

However, the M.P.E.P. sets forth the strict legal standard for establishing a *prima facie* case of obviousness based on modification of prior art references. The standard for such a modification is discussed above in Section II.A. Briefly, however, Appellants note that "[o]bviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either explicitly or implicitly in the references themselves or in the knowledge generally available to one of ordinary skill in the art" at the time of the invention." M.P.E.P. § 2143.01. Even the fact that references *can* be modified does not render the resultant modification or combination obvious unless the prior art teaches or suggests the desirability of the modification or combination. *See Id.* (citations omitted). According to the Federal Circuit, "a showing of a suggestion, teaching, or motivation to combine or modify prior art references is an essential component of an obviousness holding." *In re Sang-Su Lee*, 277 F.3d 1338, 1343, 61 U.S.P.Q.2d 1430, 1433 (Fed. Cir. 2002) (quoting *Brown & Williamson Tobacco Corp. v. Philip Morris Inc.*, 229 F.3d 1120, 1124-25, 56 U.S.P.Q.2d 1456, 1459 (Fed. Cir. 2000)). Even a determination that it would have been obvious to one of ordinary skill in the art at the time of the invention to try the proposed modification or combination is not sufficient to establish a *prima facie* case of obviousness. *See In re Fine*, 837 F.2d 1071, 1075, 5 U.S.P.Q.2d 1596, 1599 (Fed. Cir. 1988).

In the Final Office Action, the Examiner concludes that "the Newton-Raphson algorithm and Modified Gram-Schmidt algorithms are old and well known in the art" and that "it would have been obvious to one having ordinary skill in the art at the time the invention was made to include Newton-Raphson algorithm or Modified Gram-Schmidt algorithm in *Meystel*." (*Final Office Action*, page 7). As motivation for the proposed modification, the Examiner speculates that such a modification would make *Meystel* "more efficient and robust." (*Final Office Action*, page 7). However, the Examiner has not cited any evidence of

a teaching, suggestion, or motivation to modify the teachings of *Meystel*. Instead, the Examiner has merely stated that the teachings of one reference would improve the teachings of another reference. The Examiner's summary conclusion amounts to mere speculation and does not provide the suggestion or motivation necessary to make the proposed combination. The mere possibility that a modification might improve *Meystel*, as the Examiner asserts, does not even remotely provide the required teaching, suggestion, or motivation to modify the teachings of *Meystel*. Rather, this statement represents the subjective belief of the Examiner, does not point to any known authority, and therefore is not based on objective evidence of record.

It appears that the Examiner has merely proposed alleged advantages of modifying *Meystel* (advantages which Appellants do not admit could even be achieved by modifying *Meystel* in the manner the Examiner proposes). It is not sufficient to propose a modification to *Meystel* based on the mere possibility that the modification might improve *Meystel*. In other words, the advantages provided by the Examiner do not provide an explanation as to: (1) why it would have been obvious to one of ordinary skill in the art at the time of Applicants' invention (***without using Applicants' claims as a guide***) to modify the particular techniques disclosed in *Masumoto* with the cited disclosures in *Ng* and *Hsieh*; (2) how one of ordinary skill in the art at the time of Applicants' invention would have actually done so; and (3) how doing so would purportedly meet the limitations of the claims. Indeed, if it were sufficient for Examiners to merely point to a purported advantage of one reference and conclude that it would have been obvious to combine of modify that reference with other references simply based on that advantage (which, as should be evident from the case law discussed above, it certainly is not), then virtually any two or more references would be combinable just based on the fact the one reference states an advantage of its system. The Federal Circuit has made clear that this is not the law.

For at least these reasons, Appellants respectfully submit that the proposed modification of *Meystel* is improper with respect to Appellants' Claims 10 and 21. Accordingly, the rejection of Appellants' claims over the proposed *Meystel-Johnston* combination should be reversed by the Board.

CONCLUSION

Appellants have demonstrated that the present invention, as claimed, is clearly distinguishable over the prior art cited by the Examiner. Therefore, Appellants respectfully request the Board to reverse the final rejections and instruct the Examiner to issue a Notice of Allowance with respect to all pending claims.

Appellants believe that no fees are due; however, the Commissioner is hereby authorized to charge any fees or credit any overpayment to Deposit Account No. 02-0384 of Baker Botts, L.L.P.

Respectfully submitted,

BAKER BOTTS L.L.P.
Attorneys for Appellants

Jenni R. Moen

Jenni R. Moen
Reg. No. 52,038
(214) 953-6809

Date: June 12, 2007

Correspondence Address:

Customer No. **46629**

Claims Appendix

IN THE CLAIMS:

1. (Previously Presented) A method for integrated decision support, comprising the steps of:

receiving a plurality of decision inputs;

converting a first plurality of said received decision inputs to a plurality of graph representations;

converting a second plurality of said received decision inputs to a plurality of mathematical representations;

decomposing said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems;

detecting a plurality of strongly-connected components associated with said plurality of sub-problems, each of said plurality of strongly-connected components representing a connection between at least two of said plurality of sub-problems; and

solving said plurality of sub-problems.

2. (Original) The method of Claim 1, wherein the decomposing step further comprises the steps of:

performing dependency propagation for said plurality of sub-problems; and

placing said plurality of sub-problems in at least one predefined order for solution.

3. (Original) The method of Claim 1, wherein the detecting step comprises executing a graph-theoretic algorithm for a plurality of mathematical equations associated with said plurality of strongly-connected components to prevent over-constraining.

4. (Original) The method of Claim 1, wherein the decomposing step comprises decomposing said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of mathematical equations and algebraically solvable graph components.

5. (Previously Presented) The method of Claim 1, wherein the detecting step comprises detecting a plurality of dependency relations within said plurality of sub-problems.

6. (Previously Presented) The method of Claim 1, wherein the detecting step comprises identifying a plurality of simultaneous equations within said plurality of sub-problems.

7. (Original) The method of Claim 1, wherein the solving step comprises solving a plurality of numerical sub-problems and a plurality of algebraic sub-problems.

8. (Original) The method of Claim 1, wherein said solving step comprises:

solving a plurality of numerical relations sub-problems with a numerical solution algorithm;

solving a plurality of geometric relations sub-problems with an algebraic solution algorithm; and

solving a plurality of logical relations sub-problems with a logical inference solution algorithm.

9. (Original) The method of Claim 1, wherein said plurality of decision inputs comprises at least one of:

a plurality of option selection parameters;

a plurality of equality relation parameters;

a plurality of dependency parameters;

a plurality of production rule parameters;

a plurality of logical relation parameters;

a plurality of inequality expression parameters; and

a plurality of geometric constraint parameters.

10. (Original) The method of Claim 1, wherein the solving step comprises solving a plurality of simultaneous equations with a Newton-Raphson algorithm or Modified Gram-Schmidt algorithm.

11. (Previously Presented) Software for integrated decision support, the software being embodied in a computer-readable medium and when executed operable to:

receive a plurality of decision inputs;

convert a first plurality of said received decision inputs to a plurality of graph representations;

convert a second plurality of said received decision inputs to a plurality of mathematical representations;

decompose said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems;

detect a plurality of strongly-connected components associated with said plurality of sub-problems, each of said plurality of strongly-connected components representing a connection between at least two of said plurality of sub-problems; and

solve said plurality of sub-problems.

12. (Previously Presented) A computer-implemented system for integrated decision support, comprising:

a processor; and

a data storage device coupled to said processor, said processor operable to:

receive a plurality of decision inputs;

convert a first plurality of said received decision inputs to a plurality of graph representations;

convert a second plurality of said received decision inputs to a plurality of mathematical representations;

decompose said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems;

detect a plurality of strongly-connected components associated with said plurality of sub-problems, each of said plurality of strongly-connected components representing a connection between at least two of said plurality of sub-problems; and

solve said plurality of sub-problems.

13. (Original) The system of Claim 12, wherein said processor is further operable to:

perform dependency propagation for said plurality of sub-problems; and

place said plurality of sub-problems in at least one predefined order for solution.

14. (Original) The system of Claim 12, wherein said processor is further operable to execute a graph-theoretic algorithm for a plurality of mathematical equations associated with said plurality of strongly-connected components to prevent over-constraining.

15. (Original) The system of Claim 12, wherein said processor is further operable to decompose said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of mathematical equations and algebraically solvable graph components.

16. (Previously Presented) The system of Claim 12, wherein said processor is further operable to detect a plurality of dependency relations within said plurality of sub-problems.

17. (Previously Presented) The system of Claim 12, wherein said processor is further operable to identify a plurality of simultaneous equations within said plurality of sub-problems.

18. (Original) The system of Claim 12, wherein said processor is further operable to solve a plurality of numerical sub-problems and a plurality of algebraic sub-problems.

19. (Original) The system of Claim 12, wherein said processor is further operable to:

solve a plurality of numerical relations sub-problems with a numerical solution algorithm;

solve a plurality of geometric relations sub-problems with an algebraic solution algorithm; and

solve a plurality of logical relations sub-problems with a logical inference solution algorithm.

20. (Original) The system of Claim 12, wherein said plurality of decision inputs comprises at least one of:

a plurality of option selection parameters;

a plurality of equality relation parameters;

a plurality of dependency parameters;

a plurality of production rule parameters;

a plurality of logical relation parameters;

a plurality of inequality expression parameters; and

a plurality of geometric constraint parameters.

21. (Original) The system of Claim 12, wherein said processor is further operable to solve a plurality of simultaneous equations with a Newton-Raphson algorithm or Modified Gram-Schmidt algorithm.

22. (Previously Presented) A system for integrated decision support, comprising:

means for receiving a plurality of decision inputs;

means for converting a first plurality of said received decision inputs to a plurality of graph representations;

means for converting a second plurality of said received decision inputs to a plurality of mathematical representations;

means for decomposing said converted first plurality of said received decision inputs and said converted second plurality of said received decision inputs to a plurality of sub-problems;

means for detecting a plurality of strongly-connected components associated with said plurality of sub-problems, each of said plurality of strongly-connected components representing a connection between at least two of said plurality of sub-problems; and

means for solving said plurality of sub-problems.

Evidence Appendix

Evidence Appendix 1

Meystel

US006102958A

United States Patent [19]

Meystel et al.

[11] **Patent Number: 6,102,958**

[45] **Date of Patent: *Aug. 15, 2000**

[54] **MULTIRESOLUTIONAL DECISION SUPPORT SYSTEM**

[75] Inventors: **Alexander Meystel**, Bala Cynwyd, Pa.; **Sameh Uzzaman**, Cherry Hill, N.J.

[73] Assignee: **Drexel University**, Philadelphia, Pa.

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: **08/835,539**

[22] Filed: **Apr. 8, 1997**

[51] **Int. Cl.**[7] **G06F 17/10**; G06F 101/10

[52] **U.S. Cl.** **703/2**; 703/1; 703/7; 703/12; 700/286; 706/906; 706/907

[58] **Field of Search** 706/906, 907; 703/1, 2, 7, 9, 12; 700/286

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,083,256	1/1992	Trovato et al.	700/56
5,301,284	4/1994	Estes et al.	711/203
5,305,230	4/1994	Matsumoto et al.	364/528.22

OTHER PUBLICATIONS

Karsai et al, "Models and Model Building In Intelligent Process Control Systems", 5th IEEE International Symposium on Intelligent Control, pp. 351–357, Sep. 1990.

"Semiotic Multiresolutional Analysis of a Power Plant"; C. Corson, A. Meystel, F. Otsu and S. Uzzaman; Aug. 27–29, 1995.

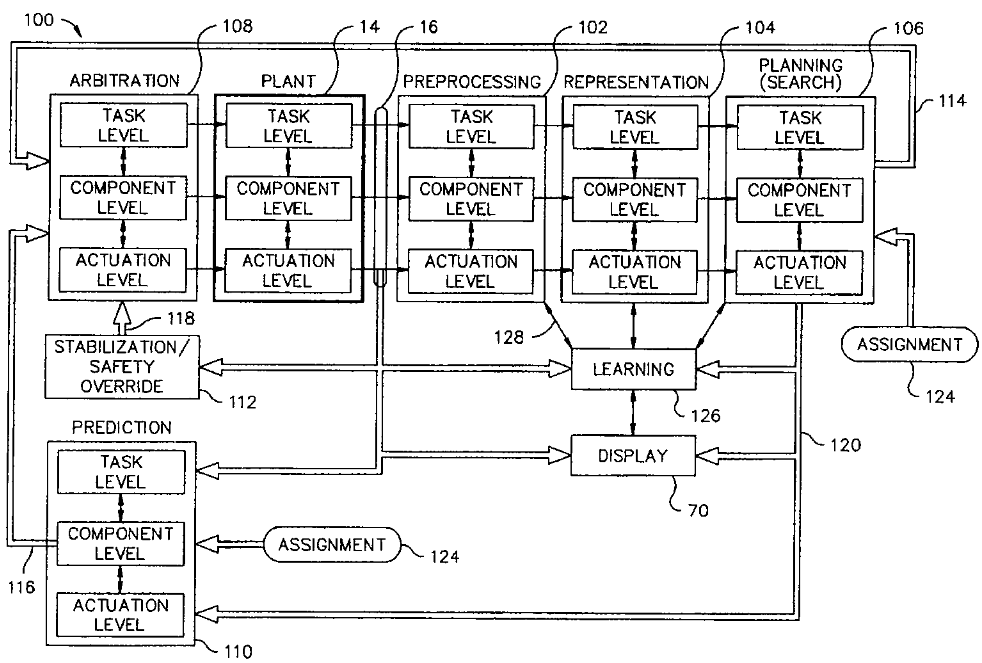
"System with Multiscale Architectures for Representation and Transformation (SMART) Plant Project—Phase I Edge Moor Demonstration—Final Report"; F. Otsu, C. Corson, A. Meystel, S. Uzzaman, H. Ward and J. Dougherty, Jun. 30, 1995.

Primary Examiner—Kevin J. Teska
Assistant Examiner—Samuel Broda
Attorney, Agent, or Firm—Akin, Gump, Strauss, Hauer & Feld, L.L.P.

[57] **ABSTRACT**

A process control system determines optimal trajectories (input controls) using multiresolutional analysis of acquired data. In contrast to conventional control systems, the present control system does not use a predetermined mathematical model or algorithm to define the process in terms of a plurality of variables. Rather, the present system acquires system data and stores the data in a multiresolutional data structure. A knowledge base is created which can be searched at varying levels of resolution for determining optimal process trajectories. The continual addition of data to the data structure allows for continual top-down refinement of the determined trajectories and bottom-up improvement and updating of the system representation.

5 Claims, 10 Drawing Sheets



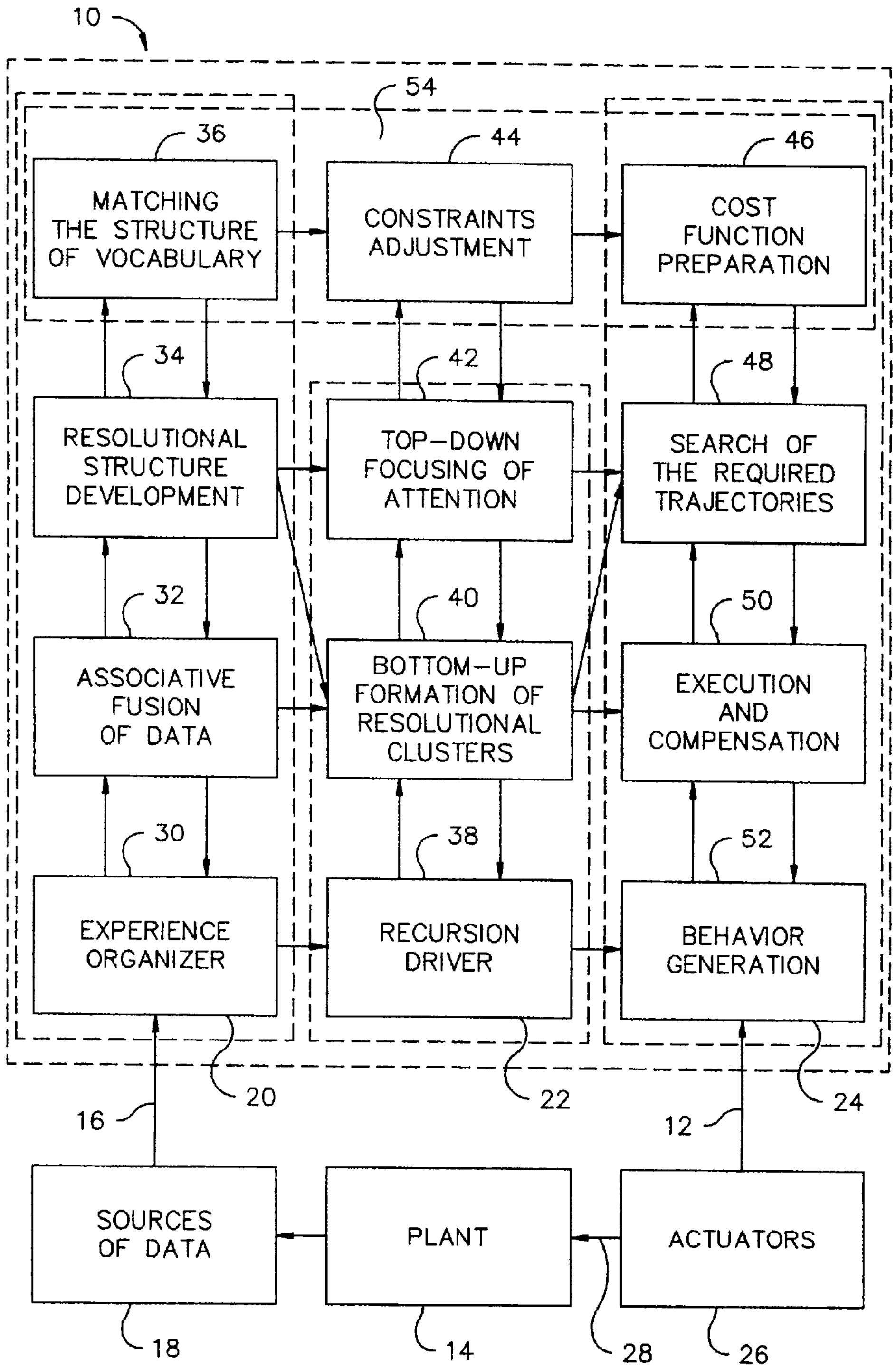

10
54
36
MATCHING THE STRUCTURE OF VOCABULARY
44
CONSTRAINTS ADJUSTMENT
46
COST FUNCTION PREPARATION
34
RESOLUTIONAL STRUCTURE DEVELOPMENT
42
TOP-DOWN FOCUSING OF ATTENTION
48
SEARCH OF THE REQUIRED TRAJECTORIES
32
ASSOCIATIVE FUSION OF DATA
40
BOTTOM-UP FORMATION OF RESOLUTIONAL CLUSTERS
50
EXECUTION AND COMPENSATION
30
EXPERIENCE ORGANIZER
38
RECURSION DRIVER
52
BEHAVIOR GENERATION
20
22
24
16
12
SOURCES OF DATA
PLANT
ACTUATORS
18
14
28
26

Fig. 1

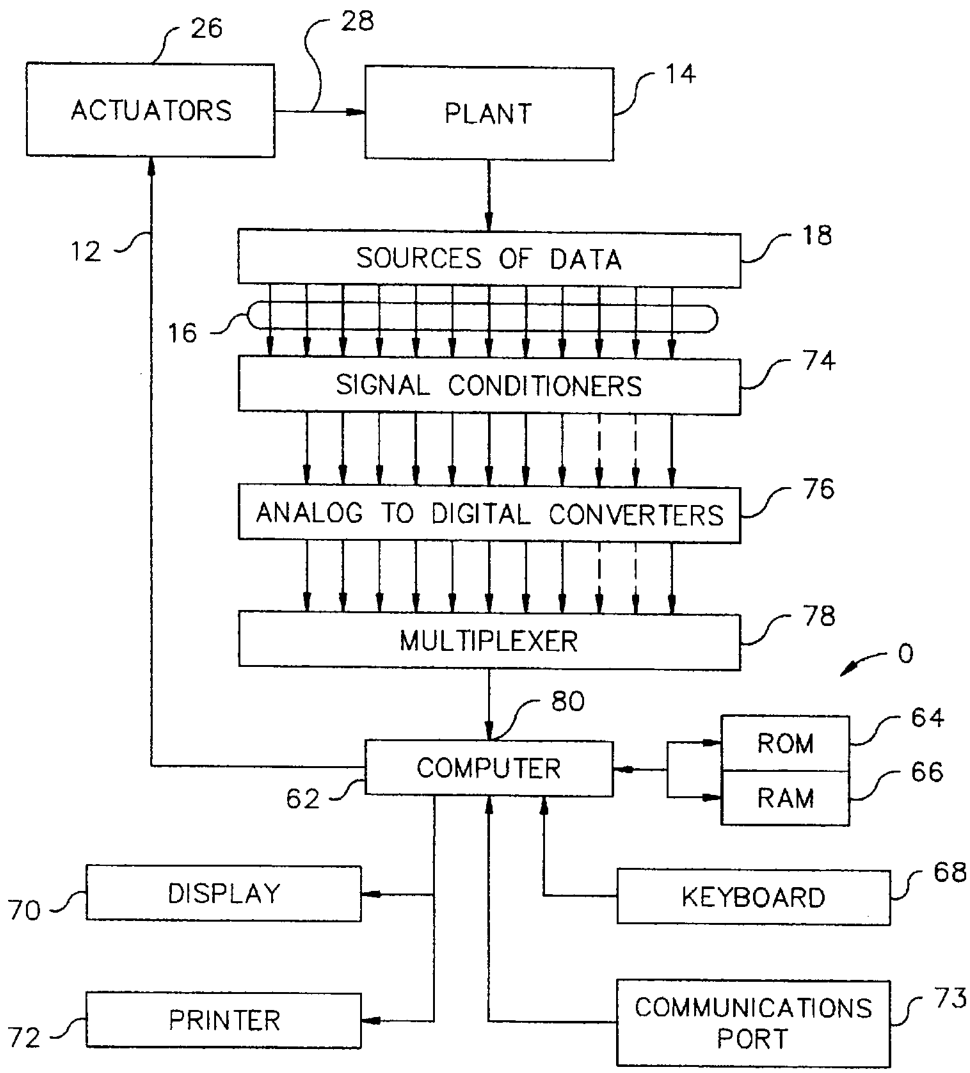

26
28
ACTUATORS
PLANT
14
12
SOURCES OF DATA
18
16
SIGNAL CONDITIONERS
74
ANALOG TO DIGITAL CONVERTERS
76
MULTIPLEXER
78
0
80
COMPUTER
62
ROM
64
RAM
66
DISPLAY
70
KEYBOARD
68
PRINTER
72
COMMUNICATIONS PORT
73

Fig. 2

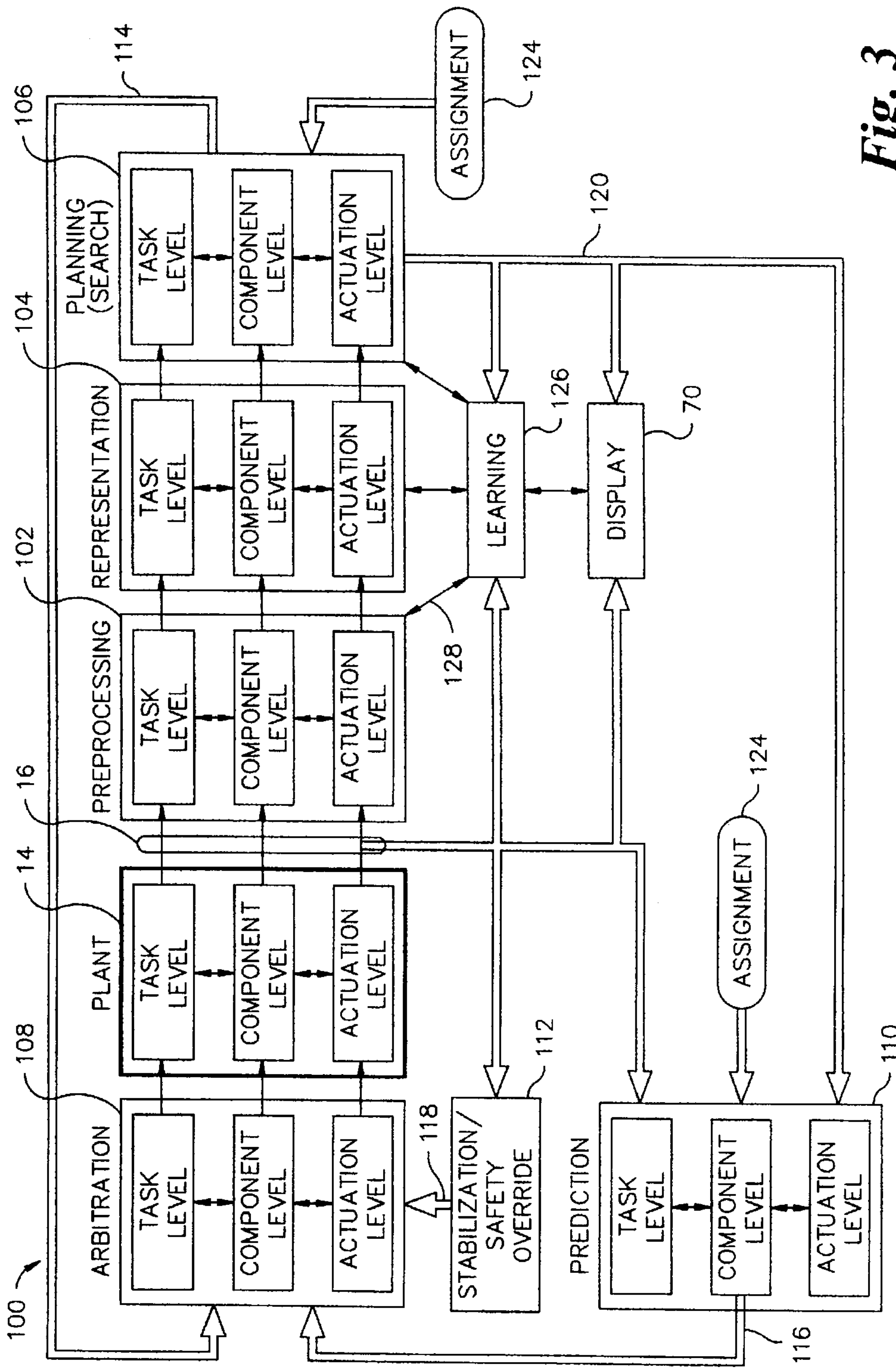

100
114
106
PLANNING (SEARCH)
TASK LEVEL
COMPONENT LEVEL
ACTUATION LEVEL
ASSIGNMENT
124
120
104
REPRESENTATION
TASK LEVEL
COMPONENT LEVEL
ACTUATION LEVEL
LEARNING
126
DISPLAY
70
102
PREPROCESSING
TASK LEVEL
COMPONENT LEVEL
ACTUATION LEVEL
128
16
14
PLANT
TASK LEVEL
COMPONENT LEVEL
ACTUATION LEVEL
ASSIGNMENT
124
108
ARBITRATION
TASK LEVEL
COMPONENT LEVEL
ACTUATION LEVEL
118
STABILIZATION/ SAFETY OVERRIDE
112
PREDICTION
TASK LEVEL
COMPONENT LEVEL
ACTUATION LEVEL
110
116

Fig. 3

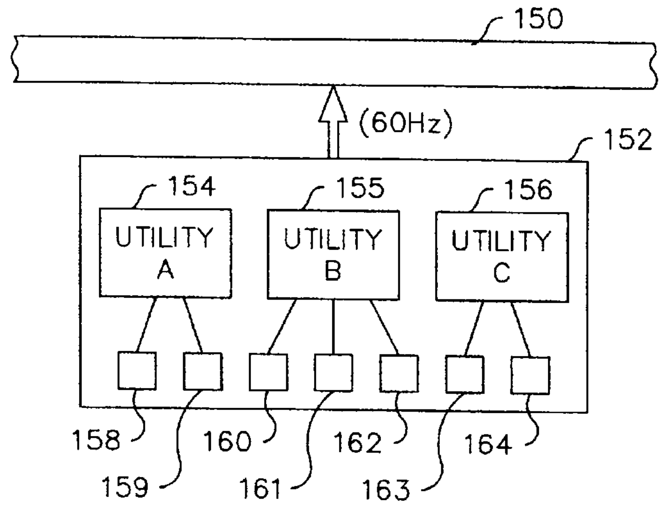

150
(60Hz)
152
154
UTILITY A
155
UTILITY B
156
UTILITY C
158
159
160
161
162
163
164

Fig. 4

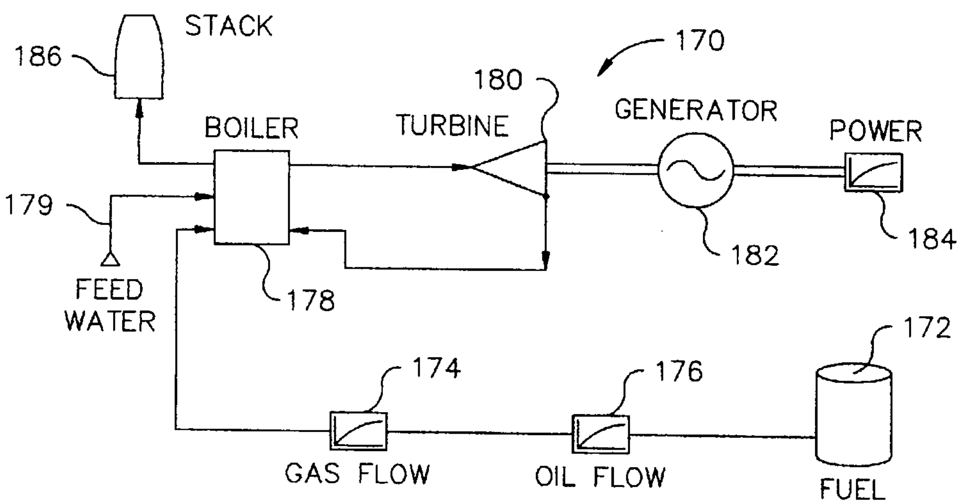

186
STACK
170
180
BOILER
TURBINE
GENERATOR
POWER
179
FEED WATER
178
182
184
172
174
176
GAS FLOW
OIL FLOW
FUEL

Fig. 5

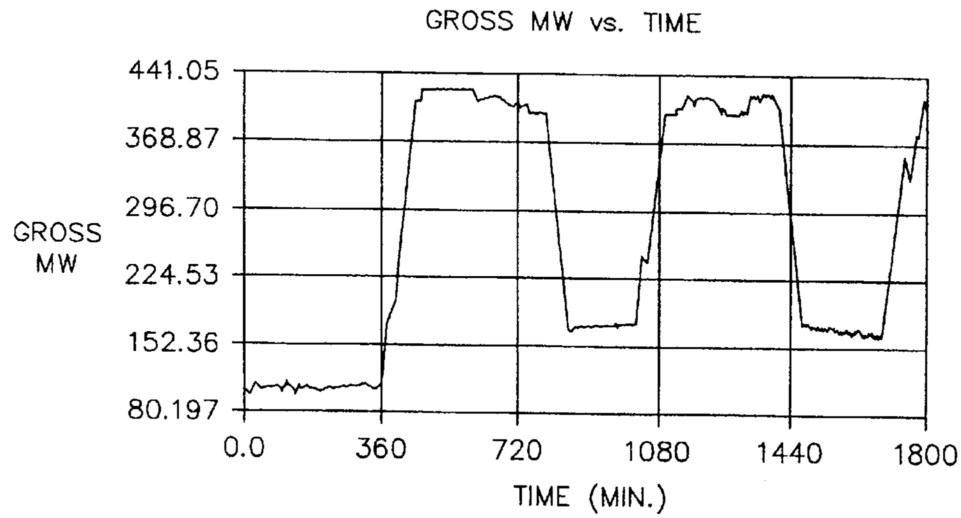

GROSS MW vs. TIME
441.05
368.87
296.70
224.53
152.36
80.197
GROSS MW
0.0
360
720
1080
1440
1800
TIME (MIN.)

Fig. 6A

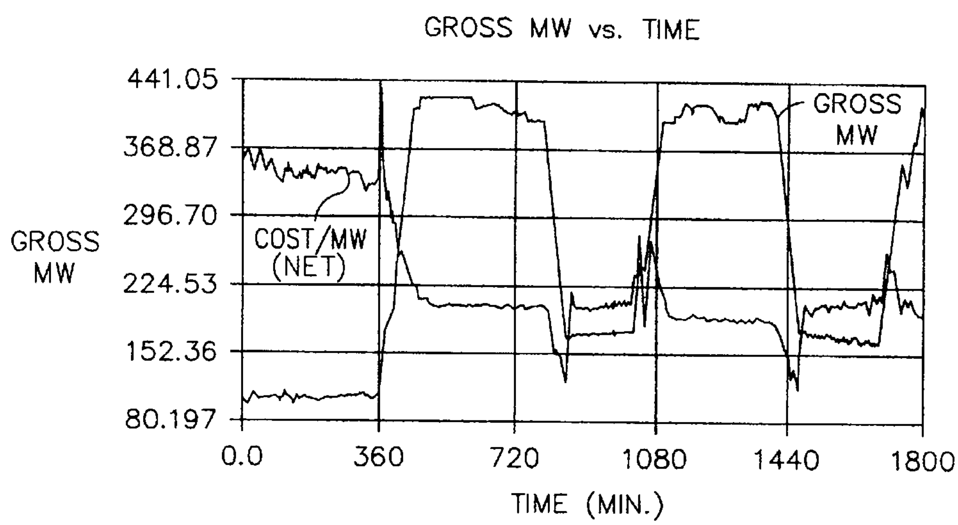

GROSS MW vs. TIME
441.05
368.87
296.70
224.53
152.36
80.197
GROSS MW
COST/MW (NET)
GROSS MW
0.0
360
720
1080
1440
1800
TIME (MIN.)

Fig. 6B

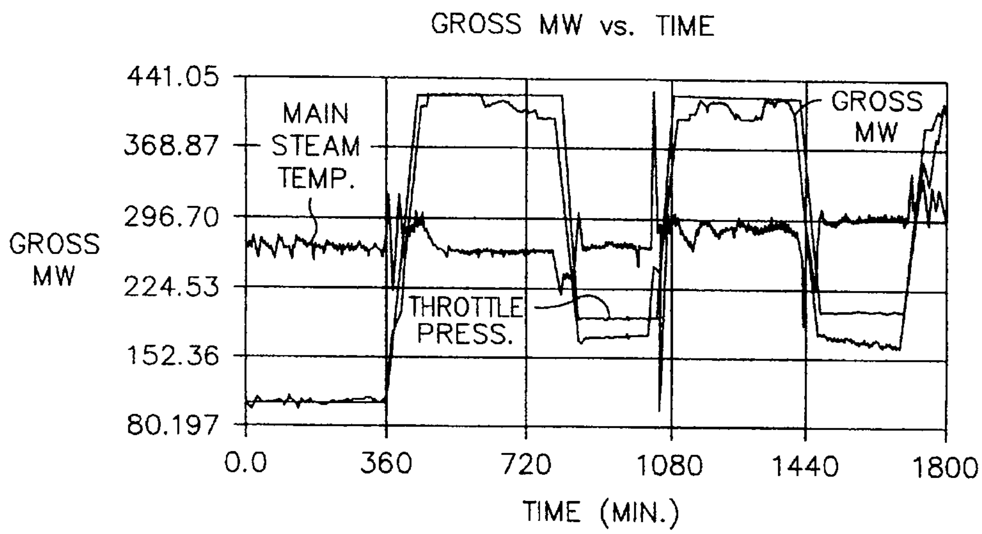

GROSS MW vs. TIME
441.05
368.87
296.70
224.53
152.36
80.197
GROSS MW
MAIN STEAM TEMP.
THROTTLE PRESS.
GROSS MW
0.0
360
720
1080
1440
1800
TIME (MIN.)

Fig. 6C

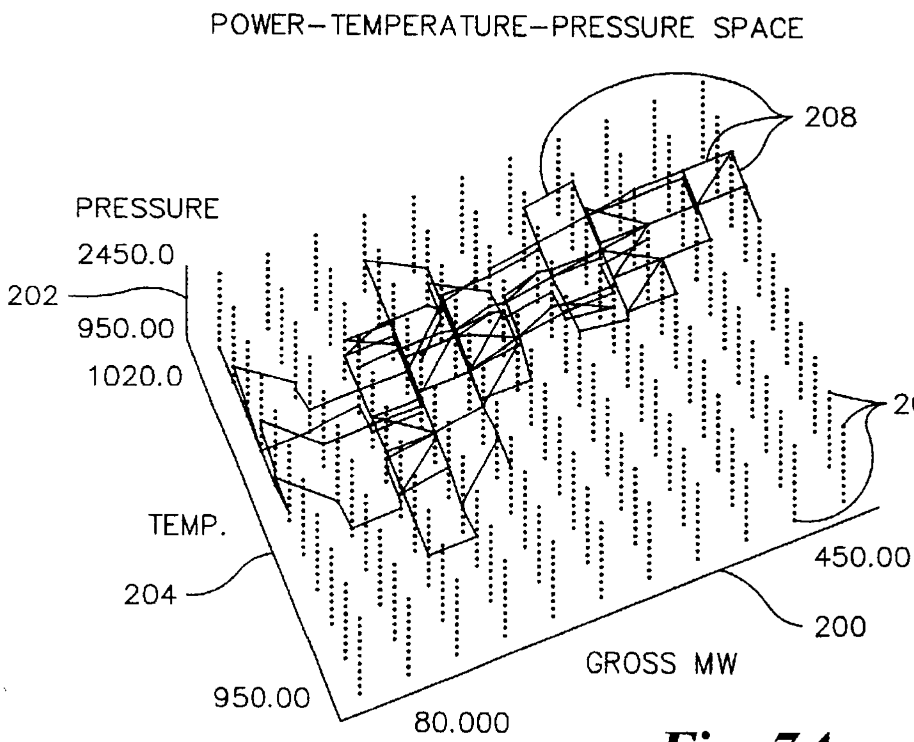
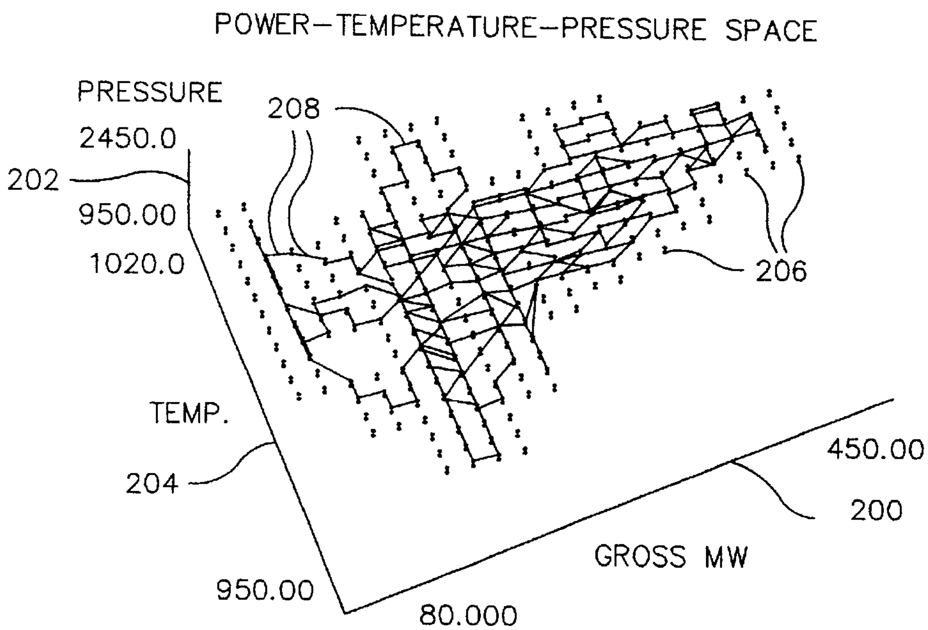
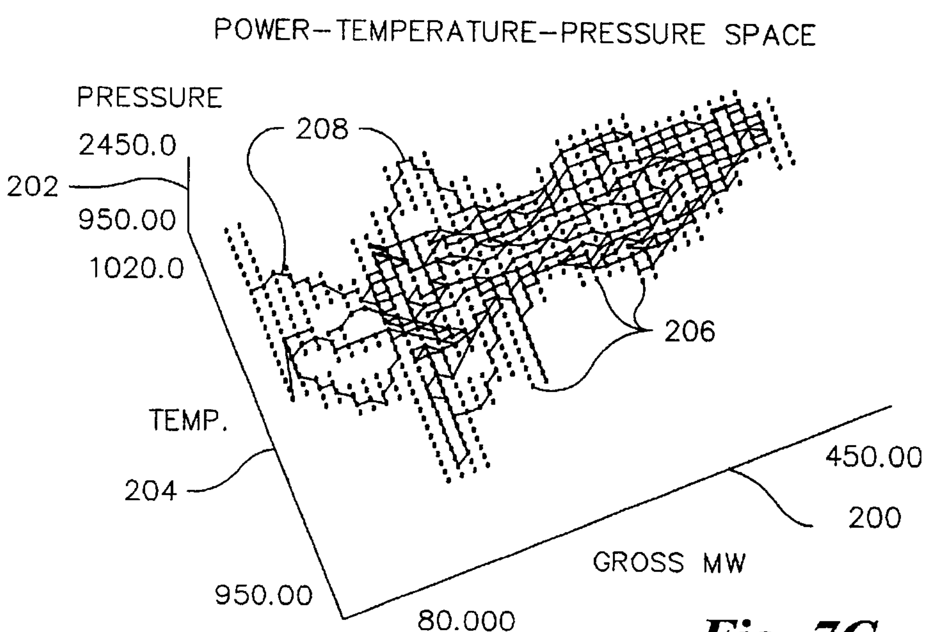

POWER-TEMPERATURE-PRESSURE SPACE
PRESSURE
2450.0
202
950.00
1020.0
TEMP.
204
950.00
80.000
GROSS MW
200
450.00
206
208

Fig. 7A



POWER–TEMPERATURE–PRESSURE SPACE
PRESSURE
2450.0
202
950.00
1020.0
208
206
TEMP.
204
950.00
80.000
GROSS MW
450.00
200

Fig. 7B



POWER–TEMPERATURE–PRESSURE SPACE
PRESSURE
2450.0
202
950.00
1020.0
208
206
TEMP.
204
950.00
80.000
GROSS MW
450.00
200

Fig. 7C

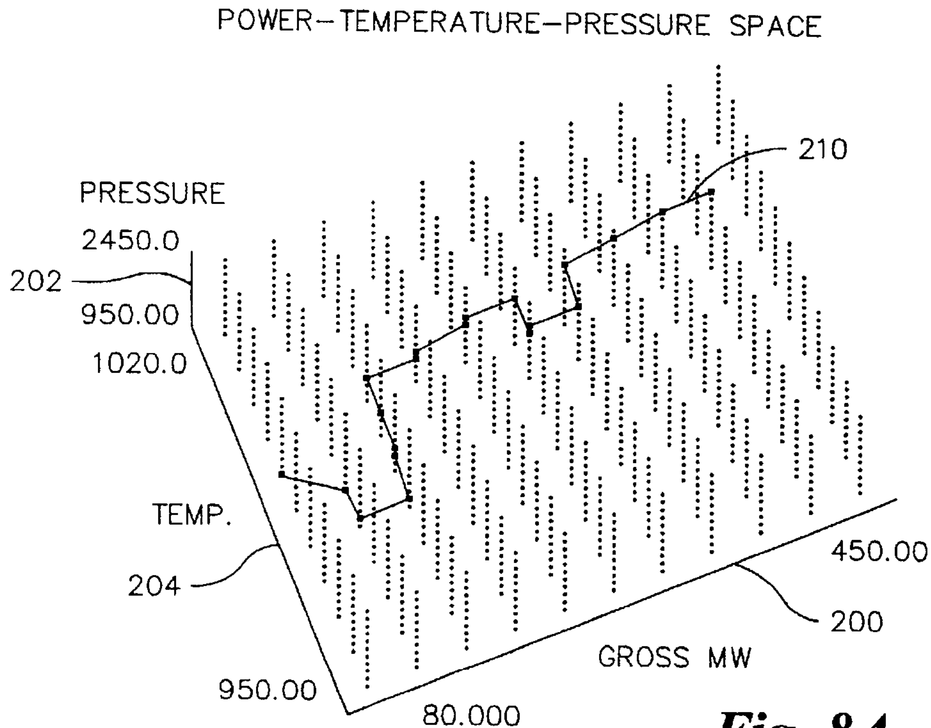

POWER–TEMPERATURE–PRESSURE SPACE
PRESSURE
2450.0
202
950.00
1020.0
210
TEMP.
204
450.00
200
GROSS MW
950.00
80.000

Fig. 8A

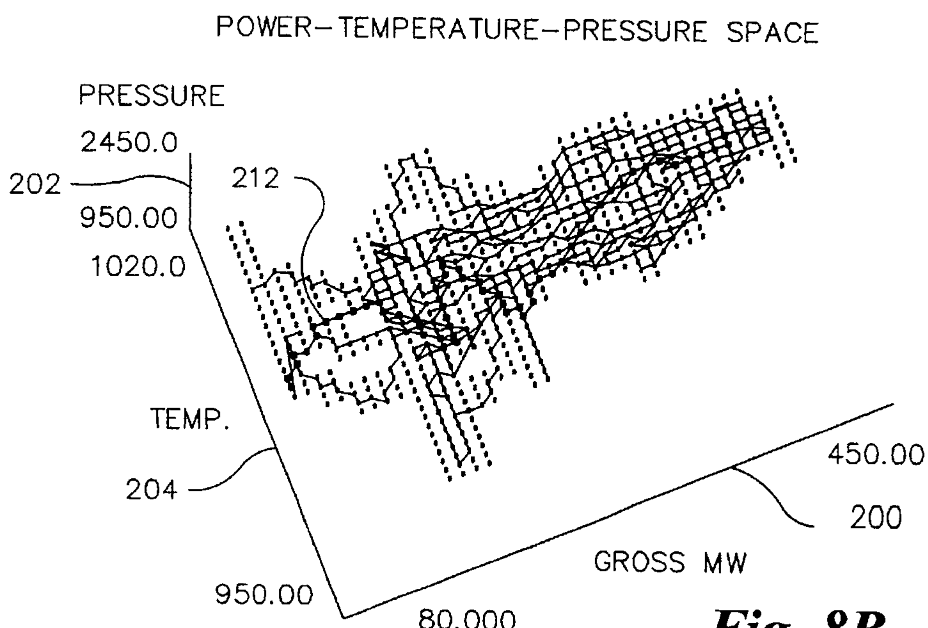

POWER–TEMPERATURE–PRESSURE SPACE
PRESSURE
2450.0
202
212
950.00
1020.0
TEMP.
204
450.00
200
GROSS MW
950.00
80.000

Fig. 8B

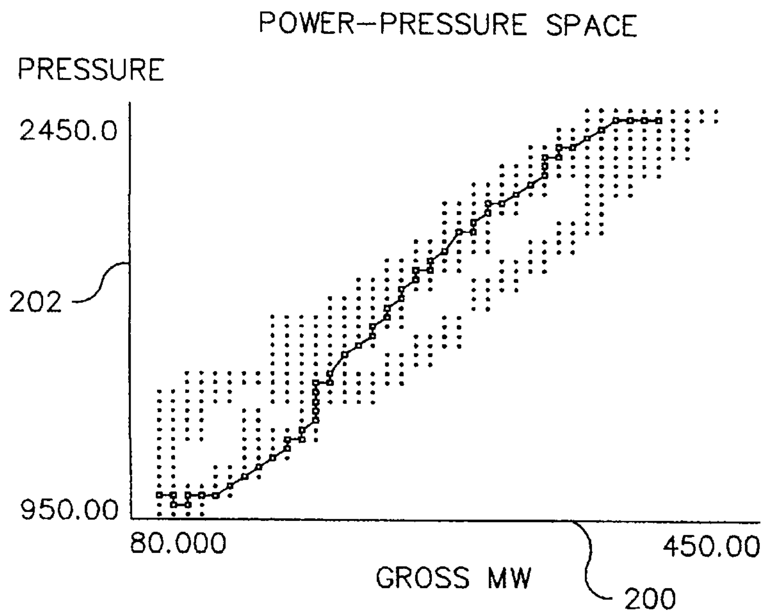

POWER–PRESSURE SPACE
PRESSURE
2450.0
202
950.00
80.000
450.00
GROSS MW
200

Fig. 9A

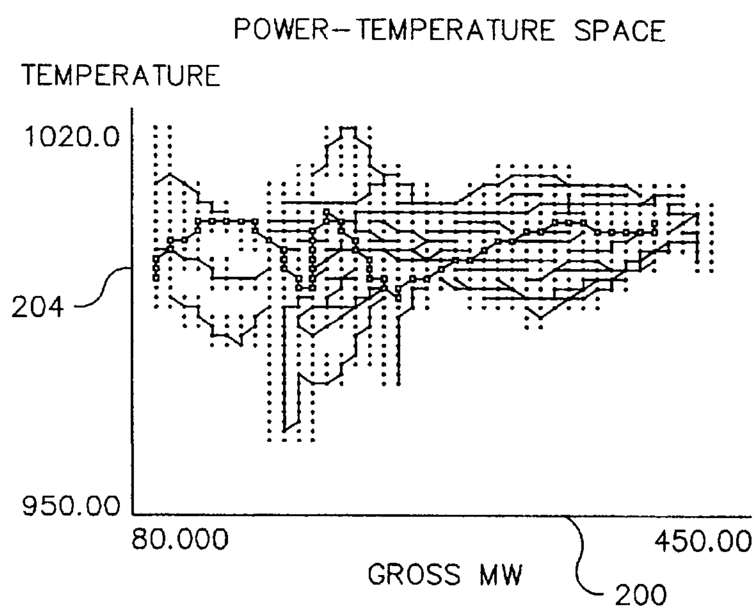

POWER–TEMPERATURE SPACE
TEMPERATURE
1020.0
204
950.00
80.000
450.00
GROSS MW
200

Fig. 9B

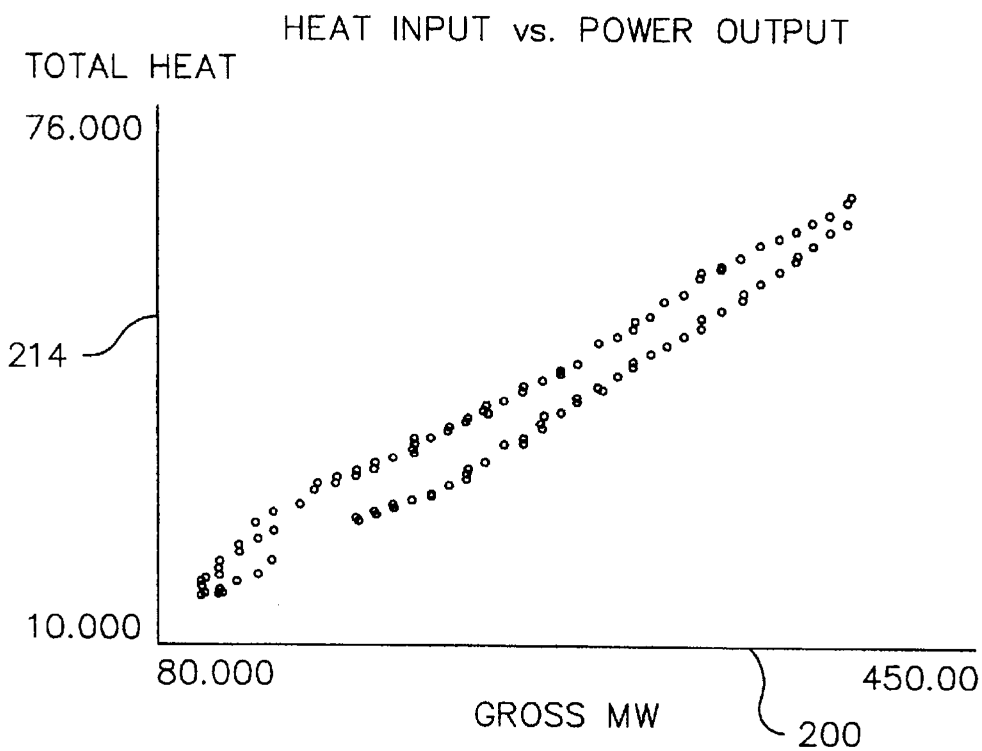

HEAT INPUT vs. POWER OUTPUT
TOTAL HEAT
76.000
214
10.000
80.000
450.00
GROSS MW
200

Fig. 10

MULTIRESOLUTIONAL DECISION SUPPORT SYSTEM

BACKGROUND OF THE INVENTION

The present invention relates to a system for multiresolutional analysis of a complex system which cannot be adequately described with a mathematical model, and more particularly, to a method of multiresolutional analysis for optimizing the performance of an industrial facility, system and/or device which do not have an adequate model of representation.

Human beings accomplish tasks by developing an organized hierarchical structuring of information which "lumps" information at various levels of "resolution" (granularity, and/or scale). The information is then retrieved and employed at the appropriate resolution level. Information collected at a more general, or overview level is generally considered to be less precise, but more easily retained and retrieved. Such data is said to be low resolution level. Information collected at a detailed level is typically more precise, but is more difficult to retain and retrieve because of its abundance. Also it increases the complexity of computation to a prohibitively high level. Such detailed information is generally termed high resolution level data. The complexity of computations at this level can be bounded by reducing the scope of the represented space of interest. When an individual acts on a task, the normal process is to constrain detailed action (higher resolution activity) based on general information (lower resolution data). A process of selecting "good" versus "bad" data is employed to make decisions.

Simulation and modeling systems use mathematical models to fully describe systems or activities. The difference between the human approach and a computer-simulated approach is that the human mind does not rely on such mathematical models. The concept and application of different resolution levels to automation and control systems is normally referred to as "supervisory control" and has been used in limited applications where the lower resolution system is only one or two levels above the higher resolution system. One such example is a unit load control system in a power plant. In such a system, the required megawatt (MW) output demand is defined at an overview level for the system. The demand value is then used in constraining functions to control subsystems (e.g., fuel/air, feed water, turbine valves) such that the megawatt demand is met. The operation of a power generating station highlights the semiotic relationships of dependent and independent process variables at multiple, discrete levels of resolution.

Generally, planning and control systems include a specified goal according to some cost or other criterion. The classical approach to designing a control system generally comprises three stages: (1) develop or identify a model of the process or operation to be controlled, (2) validate the model using a known range of operating conditions, and (3) select optimal input values to achieve a desired system behavior. The first stage, development of a model, involves the selection of an appropriate mathematical structure for purposes of identification. The mathematical structure represents a class of models which may be used to reproduce the input-output relationship of the system to be controlled. The semiotic character of the relationships in this case is taken care of by the symbol grounding procedures which employ on-line or pretabulated operations of interpretation. The mathematical structure is associated with a set of parameters, Θ, so that the class of models may be represented in symbolic form as M(Θ). The identification problem involves determining the parameter set such that the difference between an observed and predicted input-output behavior of the target system is minimized across the class of models. The second, validation stage may be performed concurrently with identification. The role of the validation stage is to verify the fidelity of the model across the expected range of operating conditions for the target system. In a system with an analytical model, validation is of deep concern because, in reality, the models are valid only for a limited time period, and in large complex systems, the limited time period is very small. The final stage is behavior generation, where having synthesized an adequate mathematical model, the selection of optimal input-output behavior is attempted using one of several existing analytical techniques. The latter are usually associated with well-known techniques (such as dynamic programming and/or maximum principle) however all of them dwell upon search algorithms and are very costly computationally. In order to provide for the input-output behavior, input-output maps are computed and a second set of maps is generated for error compensation. When this approach is applied on-line, it is called control; when this approach is applied off-line, it is called planning. The latter is used to compute feedback control strings so that the on-line control (feedback compensation) can be done only to compensate for deviations from a feed-forward generated planned trajectory.

In the classical approach, several implicit assumptions are made in order to simplify the required analysis. The most common and significant of the assumptions is that the class of models may be chosen from a set of smooth differential equations of the type:

$$y(t)+a_1y^{(n-1)}(t)+a_2y^{(n-2)}(t)+\ldots+a_ny(t)=b_1u^{(n-1)}(t)+b_2u^{(n-2)}(t)+\ldots+b_nu(t)$$

for which the parameter set could be written as $\Theta=(a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n)$. A more general formulation of the class of mathematical structures might be closer to

$$x(t)=Ax(t)+Bu(t)+V(t)+N(x,u,t)$$

$$y(t)=Cx(t)+W(t)+M(x,u,t)$$

where the A, B, and C matrices are coefficients of terms linear in state and input, V and W are noise terms, and N and M are non-linear components of the model. Further, generality could be obtained by including partial derivatives in the formulation.

The use of a class of candidate models which is capable of representing stochastic and non-linear dynamics which contribute significantly to system behavior in many settings, raises the complexity of behavior generation and optimization strategies. The difficulties grow substantially when both input-output pairs, as well as the parameter set are subject to disturbances. The problem becomes very difficult to solve when the system demonstrates distributed character. Existing analytical techniques are not suited to mathematical models in this form. Consequently, superior identification techniques are of little value unless methods of behavior generation and optimization are utilized which can determine useful strategies for this class of models. The classical approach is especially difficult to apply when trying to control a large complex system. Mathematical models for such systems are not well known and often are erroneous, so that the results of computations are fuzzy.

The present invention is directed to a process control system which determines optimal trajectories (input

controls) using multiresolutional analysis of acquired data. In contrast to prior art control systems, the present invention does not use a predetermined mathematical model or algorithm which defines the process in terms of a plurality of variables. Rather, the present invention acquires system data and stores the data in a multiresolutional data structure. A knowledge base is thus created, which is searched at varying levels of resolution for determining optimal process trajectories. This base can be called "a model" in a very general sense. It is rather a fluid "provisional data structure" which is created and recreated on demand, to support the decision making process. The continual addition of data to the data structure allows for continual refinement of the determined trajectories.

SUMMARY OF THE INVENTION

Briefly stated, the present invention comprises a multiresolutional decision support system for determining output trajectories for a predetermined output behavior of a plant or a process and for generating control signals for achieving the predetermined output behavior. The system comprises:

means for acquiring data related to at least one process variable of the plant or process;

means for organizing the acquired data into goal independent units and fusing the units into associative clusters of information;

means for storing the associative clusters of information in a multiresolutional data storage structure;

means for searching the multiresolutional storage structure at successively more abstract levels of resolution in a bottom-up processing, or less and less abstract levels of resolution in a top-down processing, both utilized to determine a trajectory for achieving the predetermined output behavior; and

means for translating the search results into control signals for causing the plant or process to achieve the predetermined output behavior.

In a second embodiment, a multiresolutional system for determining optimal trajectories of a process having a plurality of operational variables and cost functions is disclosed in which the system comprises:

means for acquiring data relating to at least one of the operational variables;

means for organizing the operational variables into a representative hierarchy;

a multiresolutional data storage structure for storing the acquired data according to the representative hierarchy;

means for embedding local dynamics of the process into at least one level of resolution of the data stored in the data storage structure; and

means for synthesizing optimal trajectories of process operation using a search applied to the multiresolutional representation.

In yet another embodiment, a multiresolutional decision support system for monitoring and controlling a plant or process and for determining optimal trajectories of operation for achieving a predefined operational behavior of the plant or process, the plant or process including a plurality of operational variables and cost constraints is disclosed in which the system comprises:

a plurality of sensors for monitoring and acquiring data from a plurality of subsystems of the plant or process, the data relating to at least one of the operational variables;

a memory device for storing the acquired data;

means for organizing the operational variables into groups of goal-independent input-output rules;

a data structure for storing the acquired data in a multiresolutional representation in accordance with the rules such that local dynamics of the plant or process are embedded in the data structure;

processor means for recursive algorithmic multiresolutional search of the data stored in the data structure at varying levels of resolution and generating an optimal trajectory of operation of the plant or process for achieving the predefined operational behavior; and

control signal generation means for generating at least one control signal for controlling the operation of the plant or process according to the optimal trajectory in order to achieve the predefined operational behavior.

A further embodiment of the present invention comprises, in a power plant, a planning and control system for determining output trajectories of a predefined output behavior and input control commands for achieving the predefined output behavior, the system comprising:

means for acquiring data related to at least one process variable of the power plant;

means for organizing the acquired data and generating at least one state input-output string therefrom;

means for fusing the acquired data into associative clusters, the clusters forming a function oriented, goal-independent provisional model of the power plant;

means for generating a multiresolutional data structure based upon the associative data clusters for transforming the provisional model into at least one level of resolution incorporating embedded local dynamics of the power plant;

means for searching the multiresolutional data structure at varying levels of resolution to determine an optimal trajectory of operation for achieving the predefined output behavior; and

means for generating control signals for controlling at least one operational variable of the power plant such that the power plant operates according to the optimal trajectory.

In yet another embodiment, the present invention comprises a multiresolutional control system for plants and processes having a plurality of operational variables, constraints, and cost-functions, the control system for determining trajectories and corresponding input control signals for achieving a predefined output behavior, the system comprising:

(a) means for acquiring and organizing information related to at least one of the variables in the form of time-tagged quadruplets "state before-action-state after-value" strings;

(b) means for extraction of goal-independent data from the goal-dependent strings in the form of generalized rules;

(c) fusing components of the rules into associative clusters which form a function oriented goal-independent provisional model of the system, wherein multiple goal-independent and time-independent rules are clustered into more general rule statements, thereby producing a hierarchical system for storage and retrieval of rules;

(d) means for developing a multiresolutional data structure based upon the associative clusters which transforms the provisional relational model of the system into a multilevel hierarchical relational structure;

(e) means for bottom-up formation of resolutional clusters which provide for further development of the provisional model of the system;

(f) means for top-down focusing of attention upon a subset of data within the provisional model and developing envelopes of attention at least one level of resolution;

(g) means for matching the predefined output behavior to the provisional model of the system for at least one level of resolution;

(h) means for adjusting the constraints of the predefined operational behavior;

(I) means for adjusting the cost function of the system to the model of at least one level of resolution with at least one envelope of attention;

(j) means for searching within at least one envelope of attention for an optimal trajectory for achieving the predefined output behavior, wherein the process of searching is repeated consecutively at one or more levels of resolution with narrowing the search envelope and consecutive refinement of the results; and

(k) means for generating system control signals at least at one level of resolution (otherwise, organized in a multiresolutional structure of control signals) related to the optimal trajectory for achieving the predefined system behavior.

The present invention further is a method of determining output trajectories for a predetermined output behavior of a plant or a process and for generating control signals for achieving the predetermined output behavior, the method comprising the steps of:

(a) acquiring data related to at least one process variable of the plant or process;

(b) organizing the acquired data into goal independent units and fusing the units into associated clusters of information, and recursively organizing the associated clusters of information into a new set of associated clusters of lower resolution until a lowest level of resolution is achieved;

(c) storing the associated clusters of information in a multiresolutional data storage structure;

(d) searching the multiresolutional storage structure at successively higher levels of resolution to determine a trajectory for achieving the predetermined output behavior and propagating the search results successively at the lower levels of resolution; and

(e) translating the search results into control signals for causing the plant or process to achieve the predetermined output behavior.

In another embodiment, the present invention is a method of determining optimal trajectories of a process having a plurality of operational variables and cost functions, the method comprising the steps of:

(a) acquiring data relating to at least one of the operational variables;

(b) organizing the operational variables into a representative multiresolutional hierarchy;

(c) storing the acquired data according to the representative multiresolutional hierarchy in a multiresolutional data storage structure;

(d) embedding local dynamics of the process into at least one level of resolution of the data stored in the data storage structure; and

(e) synthesizing optimal trajectories of process operation using a search applied to the multiresolutional representation at successively higher levels of resolution with subsequent propagation of the search results to successively lower levels of resolution.

In yet another embodiment, the present invention is a method of monitoring and controlling a plant or process and of determining optimal trajectories of operation for achieving a predefined operational behavior of the plant or process, the plant or process including a plurality of operational variables and cost constraints, the method comprising the steps of:

(a) monitoring and acquiring data using a plurality of sensors, from a plurality of subsystems of the plant or process, the data relating to at least one of the operational variables;

(b) storing the acquired data in a memory device;

(c) organizing the operational variables into groups of goal-independent input-output rules;

(d) storing the acquired data in a data structure in a multiresolutional representation in accordance with the rules such that local dynamics of the plant or process are embedded in the data structure;

(e) recursively searching the data stored in the data structure at varying levels of resolution and synthesizing an optimal trajectory of operation of the plant or process for achieving the predefined operational behavior; and

(f) generating at least one control signal for controlling the operation of the plant or process according to the optimal trajectory in order to achieve the predefined operational behavior.

The present invention is further a method for determining output trajectories of a predefined output behavior and input control commands for achieving the predefined output behavior, in a power plant. The method comprises the steps of:

(a) acquiring data related to at least one process variable of the power plant;

(b) organizing the acquired data and generating at least one state input-output string therefrom;

(c) fusing the strings into associative clusters, the clusters forming a function oriented, goal-independent provisional model of the power plant;

(d) generating a multiresolutional data structure based upon the associative data clusters for transforming the provisional model into at least one level of resolution incorporating embedded local dynamics of the power plant;

(e) recursively searching the multiresolutional data structure at varying levels of resolution to determine an optimal trajectory of operation for achieving the predefined output behavior; and

(f) generating control signals for controlling at least one operational variable of the power plant such that the power plant operates according to the optimal trajectory.

The present invention is also a method for plants and processes having a plurality of operational variables, constraints, and cost-functions, for determining trajectories and corresponding input control signals for achieving a predefined output behavior, the method comprising the steps of:

(a) acquiring and organizing information related to at least one of the variables in the form of time-tagged quadruplets "state before-action-state after-value" strings;

(b) extracting goal-independent data from the goal-dependent strings in the form of generalized rules;

(c) fusing components of the rules into associative clusters which form a function oriented goal-independent

provisional model of the system, wherein multiple goal-independent and time-independent rules are clustered into more general rule statements, thereby producing a hierarchical system for storage and retrieval of rules;

(d) developing a multiresolutional data structure based upon the associative clusters by using a recursive computational process which transforms the provisional relational model of the system into a multilevel hierarchical relational structure;

(e) bottom-up recursive formation of resolutional clusters within the multiresolutional data structure, which provides for further development of the provisional model of the system;

(f) top-down recursive focusing of attention upon a subset of data within the provisional model and developing envelopes of attention at least one level of resolution;

(g) matching the predefined output behavior to the provisional model of the system for at least one level of resolution;

(h) adjusting the constraints of the predefined operational behavior;

(I) adjusting the cost function of the system to the model with at least one envelope of attention;

(j) searching within at least one envelope of attention for an optimal trajectory for achieving the predefined output behavior, wherein the process of searching is repeated consecutively at one or more levels of resolution with narrowing the search envelope and consecutive refinement of the results, wherein the search results are consecutively propagated from the bottom up to the lower levels of resolution; and

(k) generating system control signals related to the optimal trajectory for achieving the predefined system behavior.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing summary, as well as the following detailed description of a preferred embodiment of the invention, will be better understood when read in conjunction with the appended drawings. For the purpose of illustrating the invention, there is shown in the drawings an embodiment which is presently preferred. However, it should be understood that the invention is not limited to the precise arrangement and instrumentalities disclosed. In the drawings:

FIG. **1** is a schematic architectural block diagram of a control system in accordance with the present invention;

FIG. **2** is a schematic block diagram of a computer based control system in accordance with the present invention;

FIG. **3** is a schematic block diagram of the software architecture of the control system in accordance with FIG. **2**;

FIG. **4** is a schematic block diagram of a power generation company which supplies power to a regional power grid;

FIG. **5** is a schematic block diagram of a power generation unit;

FIG. **6**A is a graphic illustration of the relationship between output power and time for a power generation unit;

FIG. **6**B is a graphic illustration of the relationship between output power against time (FIG. **6**A) overlaid on a corresponding generating cost in terms of BTU/MW;

FIG. **6**C is a graphic illustration of the relationship between output power against time overlaid on generating cost (FIG. **6**B) shown with the accumulation of selected transients from acquired raw data;

FIGS. **7**A–**7**C are graphic illustrations of a three dimensional state space at varying levels of resolution;

FIGS. **8**A–**8**B are graphic illustrations of search results for an optimal operating trajectory of the state space of FIGS. 7A and 7C, respectively;

FIGS. **9**A–**9**B are two-dimensional graphic illustrations of FIGS. **8**A–**8**B (power against pressure and temperature, respectively); and

FIG. **10** is a graphic illustration of heat against power.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The list of desirable properties of an intelligent system includes the ability to "dynamically navigate" through an unstructured and partially unknown environment (processed state space) using a knowledge base and planning, to make predictions or form hypotheses, and to demonstrate an ability to learn. According to the present invention, it is possible to describe a dynamic system without using a conventional dynamic model, but rather by a properly organized knowledge base of prior experiences which serve as a virtual, provisional model. A recursive algorithmic process of planning, followed by the selection of a smaller space for more detailed analysis, is applied at each level of precision to derive trajectories or paths at a predefined final precision. The trajectory derivation is done in a manner which restricts the quantity of data required at each iteration, even as the working precision increases by several orders of magnitude.

The present invention comprises a multiresolutional decision support system (MDSS) for plant performance enhancement. MDSS implements selected features of multiresolutional control theory to assist in the process of making optimal, or cost minimizing, decisions for both process control and design. MDSS is able to integrate information from diverse sources or subsystems and, without using models or equations, to formulate plans or rules of system operation which provide the best performance according to a set of criteria of interest, subject to the constraints of available information. One of the main features of MDSS is an ability to generalize precise information (acquired data) into successively more abstract levels of resolution, which allows a significant reduction in the complexity of analysis of alternatives. Such analysis can, therefore, be performed via successive refinement of increasingly accurate projections within a more constrained envelope at each level of resolution. Once the multiresolutional structure of consecutively generalized data is constructed, a top-down "navigation" is possible which constitutes the core of the MDSS functioning.

MDSS efficiently analyzes distributed system information obtained as a result of on-line data acquisition. Acquired data is organized in a hierarchy of scale and scope, permitting examination in a narrower envelope at each succeeding level of higher resolution. Each plan determined at a higher resolution is more detailed than at preceding (lower resolution) levels and is based on a more localized and precise examination of selected information. The hierarchical organization dramatically reduces the complexity of information storage and analysis. MDSS allows the use of a distributed modeling technique with information acquired and embedded into the model, to synthesize optimal strategies of operation.

The multiresolutional knowledge representation forms the basis that is used to synthesize feed-forward trajectories of optimal control. The collection of data is performed by testing a process or plant within the boundaries of its normal

operation. The benefits of the present invention include a reduction in the real costs of operating or designing processes, with a relatively insignificant expenditure of time and effort in the automated analysis of available system information.

MDSS allows at least three modes of operation: off-line, on-line, and design phase. Off-line mode provides supervision and planning of process information. On-line mode provides monitoring and control of processes according to predefined criteria of optimality. Design phase mode provides selection of system or process components which reduce the cost of achieving required process performance targets.

Referring now to the drawings, wherein the same reference numeral designations are applied to corresponding elements throughout the several figures, there is shown in FIG. 1 a schematic architectural block diagram of a multiresolutional decision support system **10** of the present invention for generating input commands or control commands **12** for complex systems, e.g. a manufacturing facility, a robot, a power plant or corresponding processes **14**. The control commands **12** are generated based on a calculation by the system **10** of an optimal trajectory or path for achieving a predetermined, desired output behavior of the plant or process **14**. The data **16** is input to the system **10** from sources of data **18**. The sources of data **18** monitor a plurality of variables of functioning constraints and cost-functions of the plant or process **14**. The sources of data **18** may comprise sensors (not shown) which monitor the operation of subsystems of the plant or process **14**. The sources of data **18** may be interpreted also, as humans observing the process and submitting the required information. In FIG. **1**, arrows connecting certain blocks are shown which represent the direction and flow of information between such blocks.

The system **10** may be broken down into a plurality of subsystems, with each subsystem including one or more operational modules. A first organizational subsystem **20** acquires all input based information, such as the data **16**, and organizes such information for use by a modeling subsystem **22**. The modeling subsystem **22** determines functional relationships of the organized data, which are searched by a behavior generation subsystem **24**. The behavior generation subsystem **24** generates the control commands **12**, which are provided to an actuator block **26**. The actuator block **26** translates the control commands **12** into process control signals **28** for physically controlling, modifying or altering the operation of the plant or process **14**.

The first organizational subsystem **20** preferably comprises an experience organizer module **30**, an associative fusion of data module **32**, a resolutional structure development module **34** and a matching the structure of vocabulary module **36**. The experience organizer module **30** comprises means for acquiring, organizing, and storing information, such as the data **16**, which is related to predetermined variables of interest of the plant or process **14**. Preferably, the acquired data **16** is organized into goal dependent units of information, such as time-tagged quadruplets "state-before-action, state-after-value" strings, also termed "experiences". The experiences are classified in order to permit accurate and plausible inferences to be drawn therefrom. Accordingly, the experience organizer module comprises storage structures which allow for memorizing the goal-dependent quadruplets of "experiences" and organizing their subsequent classification. Since the incoming flow of "experiences" for any process is generally very large, the volume of memory required has a tendency of growing very quickly. Thus, the storage structures for acquiring and organizing information should allow for functioning with a required level of information efficiency. The latter can be achieved by a variety of measures, such as prearranged forgetting and consecutive generalization of experiences based upon unifying experiences which are similar in preconditions, action, and effect but having a lower resolution of representation. The latter produces eventually a hierarchical system for storage and retrieval of experiences.

The associative fusion of data module **32** forms a statement of rules from the data stored and organized by the experience organizer module **30**. The rules differ in the degree of belief which is assigned based upon repetitiveness of the particular experiences (statistical) or trustworthiness of the particular source of information (expert). The associative fusion of data module **32** comprises means for extraction of goal-independent data from the stored goal-dependent "experiences", in the form of generalized rules, or components of an automata transition function and state-output functions. The extraction means may comprise any standard algorithm and/or procedure for inverting experimental causalities into the statements of rules. Preferably, more sophisticated procedures are applied to the organized data, linked with algorithms and procedures of fusing the experiences into "associative clusters". Such clusters form a function oriented, goal-independent provisional model of the plant or process **14**. In order to obtain such provisional models, a mechanism of "consecutive generalization" is applied based upon unifying groups of higher resolution information into lower resolution units. However, instead of using experiences similar in preconditions, action, and effect as inseparable units (as done by the experience organizer module **30**), the associative fusion of data module **32** looks for similarities in components of the experiences, which leads to rules generation without explicit demonstration of partial dependencies demonstrated in the acquired data **16**. Multiple goal-independent and time-independent rules are clustered into more general rule statements, thereby producing eventually a hierarchical system for storage and retrieval of rules from which the provisional relational model of the plant or process **14** is combined subsequently.

The resolutional structure development module **34** comprises means for developing a multiresolutional data (knowledge) structure based upon the "associative clusters", which can transform the provisional relational model into a multilevel hierarchical relational structure with any required number of levels of resolution, including embedded local dynamics of the plant or process **14**. Such multiresolutional data structure development means include standard algorithms and procedures for clustering and hierarchical ordering of the units forming the relational model. Although such conventional means can be used, the interpretation of the results, preferably the process of hierarchical ordering, ends when the resolutional structure of representation is obtained, wherein each level is associated with a particular resolution level of the plant or process **14**.

The matching the structure of vocabulary module **36** comprises means for matching a statement of the assignment to the content of the provisional model with at least one level of resolution. The statement of the assignment defines a desired predetermined output of the plant or process **14**. Such means for matching may comprise any standard algorithm and/or procedure for component-by-component list browsing and for ordering the lists in correspondence.

The aforedescribed modules **30–36** form a unified subsystem **20** of organization of sensory (and any input-based) information. The number of modules shown is not related to the number of the resolution levels which can be obtained by the system **10**.

The modeling subsystem **22** preferably comprises a recursion driver module **38**, a bottom-up formation of resolutional clusters module **40** and a top-down focusing of attention module **42**. The recursion driver module **38** comprises means for guiding and coordinating bottom-up and top-down recursive processes of knowledge organization performed by the bottom-up and top-down modules **40, 42**. Such guiding and coordinating means include any standard algorithm and/or procedure for registering, nesting and checking conditions of inclusion and monitoring imposition of quantitative constraints.

The bottom-up formation of resolutional clusters module **40** comprises means for bottom-up formation of multiresolutional clusters which provide for further development of the model of the plant or process **14** from a provisional model into an applied model with at least one level of resolution. Such means for bottom-up formation include all standard algorithms and procedures for clustering, including such standard solutions as K-means, as well as any another algorithmic tool of cluster generation. After completion of the cluster generation, the cluster generation algorithm is applied to the newly created clusters, which may be considered as objects, in a recursive fashion.

The top-down focusing of attention module **42** is experience-based and goal-oriented, and serves the purpose of complexity reduction. The top-down focusing of attention module **42** comprises means for top-down focusing attention upon a subset of data within the model of the plant or process **14** and developing "envelopes of attention" at each level of resolution. Such top-down focusing means may comprise any standard algorithm and/or procedure for data and knowledge pruning by imposing constraints and boundaries.

The system **10** further preferably comprises a constraints adjustment module **44**. The constraints adjustment module **44** includes means for adjusting the constraints of the assignment (desired output) under the vocabulary to the results of further development of the model over the multiresolutional model. Such adjusting means include any standard algorithm and/or procedure for checking numerical conditions for interval assignment and introducing of corrections in a case where the conditions are not satisfied.

The behavior generation subsystem **24** includes a cost function preparation module **46**, a trajectory (path) search module **48**, an execution and compensation module **50**, and a behavior generation module **52**. The cost function preparation module **46** comprises means for adjusting a cost function, formulated in the assignment (desired output), to the model with multiple envelopes of attention. Such adjusting means include any standard algorithm and/or procedure for assigning and computing a measure for a particular space (representation).

The trajectory (path) search module **48** determines a preferable measure of the space, which is preferably based both upon additive and multiplicative forms, and comprises means for searching within each envelope of attention for the assignment of the output motion and input planning/control commands, which provide for the required cost-function under the particular set of constraints. Such searching means include any standard algorithm and/or procedure for different types of search, such as depth-first, breadth-first, Dijkstra, A-star, random search, genetic search, etc., as well as techniques of dynamic programming. It will be understood by those of skill in the art that different kinds of searching may be preferable for different systems in different cases. Since the model of the plant or process **14** generally contains one or more levels of resolution, the process of searching is preferably repeated consecutively at one or more levels of resolution with narrowing the search envelope and consecutive refinement of the results if the representation contains more than one resolution level.

The execution and compensation module **50** comprises means for distributing planned sequences of control commands **12** among the actuators of the actuator block **26**, comparing a planned output with a real output, and introducing a compensation command component (i.e. a feedback signal) under a particular (previously selected) control law at each level of resolution. Such command distribution means include any standard algorithm and/or procedure of inverting the planned output to the input of an actuator and introducing a feedback law of compensation.

The behavior generation module **52** comprises means for interfacing the control commands **12** to the actuator block **26**. Such interface means include all standard algorithms and procedures for providing communication between subsystems, such as a plurality of individual signal wires, a bus structure, encoded signal generation, etc.

The matching the structure of vocabulary module **36**, the constraints adjustment module **44** and the cost function preparation module **46** may also be grouped together or unified as a second, separate, organizational subsystem **54**, since these modules **36, 44, 46** preferably function in a coordinated fashion as an organizational infrastructure of the control system **10**. The function of the second organizational system **54** is synthesis of "value judgment" required for selectors of the multiple procedures employed within the system.

Referring now to FIG. **2**, a functional schematic block diagram of a preferred embodiment of a computer-based system **60** for implementing the control system **10** of the present invention is shown. In the presently preferred embodiment, the computer system **60** includes a processing device or computer **62** which preferably is a personal computer or equivalent. The computer **62** includes a read only memory (ROM) **64** employed for storing fixed information including a computer program and random access memory (RAM) **66** of a predetermined size which is adapted for temporary storage of portions of a computer program as well as data for analysis. The computer **62** further includes a central processing unit or processor (not shown) and, in the present embodiment, a hard disk (not shown) of a type typically employed in such personal computers. While it is presently preferred that the computer **62** comprise a personal computer, it should be appreciated that any other suitable type of computer, such as a lap-top computer, mini computer, microprocessor, digital signal processor, mainframe computer, work station, etc., may alternatively be employed.

A keyboard **68** is employed as the primary input device to permit a user to communicate with the computer **62**, although other input devices may also be used instead of or in addition to the keyboard **68**, such as a mouse, a joy stick, voice commands, etc. A display, in the present embodiment a CRT display **70**, and a suitable printer **72** are connected to the computer **62** for the display and recording of output from the computer **62**. However, it should be understood that any other type of output device may be employed in addition to or instead of the display **70** and/or printer **72**. Use of the sophisticated output devices employing visualization of abstract concepts, virtual reality, etc., can make application of MDSS more convenient but does not affect its principles of operation.

The computer **62** further includes at least one communication port **73** which may be either a parallel port, a serial

port, or both. The communication port 74 is employed for receiving data from another location or sending data to another location utilizing a modem or other such transmission device (not shown) in a manner well known in the computer art.

The computer system 60 as thus far described is typical of a personal computer system well known to those skilled in the art. Variations in the input/output components of the computer system 60 may be made depending upon particular applications. For example, in some applications, it will not be necessary for both a printer 72 and a display 70. In other applications, a keyboard 68 may not be necessary. It should, therefore, be clearly understood by those skilled in the art that the present invention is not limited to the particular hardware configuration shown in FIG. 2 but may be implemented using any type of suitable hardware configuration.

The computer system 60 further includes a plurality of individual sensors shown collectively as the sources of data 18, for monitoring predetermined electrical, mechanical and other variables of the plant or process 14, and for converting the monitored characteristics into electrical signals (representing the data 16) for processing by the computer 62. In the present embodiment, the sources of data 18 includes a variety of different types of individual electrical and/or mechanical sensors of a type well known in the electrical and mechanical measurement art and commercially available from a variety of well known sources, such as vibration sensors, acoustic sensors, tachometers, pressure transducers, proximity probes, force sensors, torque sensors, temperature sensors, flow meters, voltage sensors, current sensors, etc. Complete details of the structure and operation of the various types of sensors for use with the present invention is not necessary for a complete understanding of the present invention. The particular type of sensors employed in the presently preferred embodiment should not be viewed as a limitation upon the invention. Each of the sensors are adapted to receive and convert sensed parameters related to the operation of the plant or process 14 into analogous standard electrical signals. In addition, as previously discussed, a human operator can be considered a valid source of the input information, supplementing or substituting the set of sensors.

The computer system 60 further preferably includes a plurality of signal conditioners 74 which are illustrated collectively. Preferably, a separate signal conditioner is provided for each of the sensors, with the respective output of the sensor being connected directly to the input to the respective signal conditioner 74. Each of the signal conditioners 74 functions in a manner well known in the art to amplify, impedance match, filter, scale, and otherwise standardize and improve the electrical output signal received from the corresponding sensor. Standardization of signals representing the data 16 includes conversion of currents to a proportional voltage, amplitude scaling and appropriate filtering to limit bandwidth. The precise structure and operation of each signal conditioner depends upon the particular type of sensor with which the signal conditioner 62 is employed. Preferably, each of the signal conditioners 62 further includes an anti-alias low pass filter which functions to improve the integrity of the acquired sensor data by filtering out, prior to digitizing, sensor signal frequencies greater than approximately half of the sampling rate of the digitizer.

The computer system 60 further includes a plurality of individual analog-to-digital converters 76 shown collectively. The analog-to-digital converters 76 function in a manner well known in the art to receive the conditioned and filtered analog output signals from the corresponding signal conditioner 74 and convert the received analog signals at a predetermined sampling rate into digital signals for data manipulation and analysis by the computer 62. A typical sampling rate could be 1,000 samples per second for each signal. Thus, each of the analog-to-digital converters 76 produces an output data array or bit stream corresponding to the particular sensor with which the analog-to-digital converter is associated.

The outputs of each of the analog-to-digital converters 76 are provided to the input of a multiplexer 78. The multiplexer 78 which, in the present embodiment is preferably a time division multiplexer, receives the data array signals from each of the analog-to-digital converters 76 and in a manner well known in the art transmits the received data array signals to an appropriate serial input port 80 of the computer 62 in a predetermined time spaced order.

It will be appreciated by those skilled in the art that the signal conditioners 74, analog-to-digital converters 76, and multiplexer 78 which are employed in connection with the presently preferred embodiment, are each of a type well known in the art and available from a variety of manufacturers. Complete details of the structure and operation of the signal conditioners 74, analog-to-digital converters 76, and multiplexer 78 are generally well known to those skilled in the art and need not be described in greater detail herein. Suffice it to say that the signal conditioners 74, analog-to-digital converters 76, and multiplexer 78 cooperate to take the raw operational data 16 from the sensors and convert the raw data 16 into a digital form suitable for processing by the computer 62. It will be appreciated by those skilled in the art that the functions of the signal conditioners 74, analog-to-digital converters 76, and multiplexer 78 may be combined into a single sub-assembly or may be performed in any of several different manners. Thus, while the preferred embodiment employs such components, such components are not intended to be a limitation on the present invention. Thus, the sensors, signal conditioners 74, analog-to-digital converters 76 and multiplexer 78 provide for data logging, and the computer system 60 provides for either analyzing the collected data or down-loading the collected data to another processor (not shown).

The present invention is capable of handling on-line information acquisition, disk-transfers, and manual data entry. In addition, a one-way communication link with the sensors and transmitters (not shown) can also be provided, since the operation of the system 60 can be decoupled from that of the plant or process 14. Further, the present system 60 can be integrated relatively simply with existing control systems which are capable of computer interface or contain PC level processing capabilities.

The above-described computer system 60 is employed for both on-line and off-line analysis of the operation of the plant or process 14. It should be appreciated by those skilled in the art that the present system 10 may be employed for analysis of any type of process, plant or facility whose operation is controlled by a plurality of variables, and that the embodiment shown in FIG. 2 is only for the purpose of illustrating a preferred embodiment of the structure and operation of the present invention.

Referring now to FIG. 3, a schematic block diagram of a multiresolutional intelligent control architecture in accordance with the present invention is shown. The structural hierarchy of a typical system, denoted 100, is shown with individual system elements comprising several levels of abstraction. FIG. 3 labels "task" through "actuator" levels,

in order of increasing specificity. This partitioning reflects the fact that the operating points of the system **100** are frequently expressed in terms of subsystems, subsystem components, and so on, down to individual actuators within the overall system. In order for the complete system **100** to function properly, each subsystem and all of the subsystem components and actuators must correctly follow their prescribed assignment. The system **100** uses strings of states, instead of single states, and an experiential knowledge base, instead of conventional models, as discussed in more detail below.

Generally, the system **100** comprises a preprocessing subsystem **102**, a representation subsystem **104**, a planning or search subsystem **106**, an arbitration subsystem **108**, a prediction or optimization subsystem **110**, and a stabilizing feedback subsystem **112**. The plant or process **14** being controlled is monitored by a plurality of sensors, as previously discussed, which sensors provide data **16** for use by the system **100**. The plant or process **14** includes several levels of data acquisition for a general multicomponent system running several concurrently actuated loops. The levels may be derived from physical considerations, such as dependence of one loop on a satisfactory operation of another loop. The operation of the control system **100** is discussed below.

The plant or process **14** performance can be measured by a variety of yardsticks, including financial savings, the meeting of regulatory standards, maximization of the operational life of a plant, and minimization of the need for operator intervention (reliability). As long as the cost measures, however objective, are tangible and may be evaluated from measurements, they can be incorporated into the present control system **100**, either separately or in a weighted combination. As opposed to prior art analytical approaches, there is no constraint on the simultaneous evaluation of cost by a variety of measures.

The overall strategy of optimization and successive refinement employed in MDSS is contingent upon the creation of a multiresolutional representation. The key requirements of the system **100** are that it must contain a generalized representation of system capabilities at a resolution (level) that allows global optimization without violating computational restraints and that each level must contain a more dense (precise) representation than the previous level. If this is achieved, then search at each successive level of resolution generates a more precise solution of the optimization problem. If prior experiences are informationally rich, the process of optimization is based on such rich experiences, and if prior experiences are poor, interpolation and extrapolation may be required.

A multiresolutional representation may be visualized as a hierarchical store of information, incorporating appropriately quantized descriptions of achievable system responses. The information is organized by precision into levels of increasingly detailed description. Algorithms for the incorporation of new information into the hierarchy are considered part of the system's knowledge. Thus, a finite set of information about the system **100** is assembled, which, in association with search and retrieval algorithms, answers the same questions as the more familiar mathematical abstraction or model, viz, what is the effect on measurable or identifiable plant variables of a particular excitation. Learning is implemented by collecting and integrating experiences and by subsequent random testing in the intervals between experiences. Information about the same objects is represented at several appropriate levels of resolution (nested, hierarchical, local decomposition is a complicated but apt description of this principle). The ability to analyze gross behavior, as well as local perturbations is also built in for efficient analysis of alternatives. The source of a model can be tested or carried out on any of an existing plant, an exiting dynamic model of more or less arbitrary complexity, or a custom-built model with elements that include rules, look-up tables, equations, and empirical data. Subjective and inconsistent, as well as traditional cost or penalty criteria are locally evaluated and incorporated into a model for future analysis. The overall strategy is thus one which places the burden of determining and achieving performance goals upon the control system, as opposed to a system designer. The present invention reinforces the feedback control component of a control system, using the byproduct of the planning subsystem **106** and the planning component (using learning algorithms and the feedback derived from system operation), the control system **100** is provided with the attributes of intelligence.

The major operational steps of the present invention are preprocessing and data analysis, multiresolutional model design, cost evaluation and process improvement. The first step, preprocessing and data analysis, provides an adequate description of the plant or process **14** by constructing and validating a distributed multiresolutional provisional model of the plant or process **14**. The multiresolutional model may be validated using a multi-valued graph representation, as described in more detail below. Cost evaluation may be performed using a sliding window to compute the costs of operation of the process in a tangible form. The sliding window permits the application of the model for process improvement which can then be measured in a consistent manner. The width of the sliding window is determined by the actual discretization of the available information and the required accuracy of results. The process improvement step represents the continual improvement of the process by applying search and optimization algorithms to the distributed model. Critical but previously unknown relationships of the process may be discovered during the preprocessing of information for the model.

The preprocessing subsystem **102** preprocesses the data **16** acquired from one or more sensors (not shown) throughout the plant or process **14**. The preprocessing subsystem **102** is provided for preprocessing raw plant information or data **16** using hierarchial clustering algorithms in order to synthesize a multiresolutional knowledge base. The preprocessing subsystem **102** organizes the data **16** in the form of time-tagged quadruplets [SB, A, SA, V] where SB is "state-before", A is "action", SA is "state-after" and V is "value" strings. The preprocessing subsystem **102** may comprise, for example, a storage structure which allows for storing and organizing goal-dependent quadruplets of "experiences", their subsequent classification, and for performing accurate and plausible inferences upon the results of the classification.

Process variables are identified by plant operators and managers as those being critical to successful operation of the target system, process or plant **14**. Another way of finding this set is clustering with determining existing correlations among the parameters. A set of process variables is initially defined and may be updated and expanded during the building of a model, if such additional variables are found to be necessary and are available. The process variables form what is conventionally known as a state space. The state space is bounded by the limits in each process variable and may be thought of as an n-dimensional box, where 'n' is the number of variables or axes of interest. In general, by virtue of the scope and interaction of parameters,

groups of process variables are arranged hierarchically in accordance with subsystems of the plant **14**, and are viewed as a single unit at higher levels of abstraction.

At this identification stage, the data **16** is collected at a sensor resolution for representing state trajectories in the form of strings, $S(x)=x(t+k\Delta t)$ for $k \geqq 0$. A recursive approximation is then performed which organizes the data **16** on the basis of successively more precise state transitions in the form:

$$\{\Delta X : x_1, X_2 \epsilon S(x), \text{ and } |x_1, X_2| > \epsilon\}_1$$

where ϵ_1 is a measure of the level of precision (or resolution) of the transitions. The feasible transitions are ranked on the basis of a measure of the cost of the transition which is dependent on a cost function of interest and are stored as graphs of the appropriate level of resolution. The graphs have a structure which may be described by an ordered set (V_p, V_c, C_t U_i) which may be considered as a parent vertex, a child or successor vertex, the cost of the transition from the parent to the successor, and the associated input code. The hierarchy of graphs allows the examination of previously encountered local behaviors at increasing precision in a reduced subset of the underlying states which, in turn, permits the synthesis of more and more precise strategies of operation with reduced search complexity.

Since the construction of the hierarchical store of information is predicated on appropriate multiresolutional discretization of the subset of the state space in which the system **100** operates, and since the data **16** is collected with a precision that is, in general, not the same at each particular level of the hierarchy, local interpolation is necessary to determine feasible edges in the graphs. The search for feasible edges does not require abandoning the acquired data **16**. Rather, the data **16** remains the basis point of the representation system while temporary nodes are computed so that the experiences can be adjusted to the required level of resolution.

The preprocessing subsystem **102** can be considered to perform the function of an observer, as well as a filter. Depending on the novelty or usefulness of the collected information or data **16**, the preprocessing subsystem **102** passes filtered data **16** to a learning subsystem **126** and/or to the representation subsystem **104** directly. A bidirectional arrow, denoted **128** connecting the preprocessing subsystem **102** to the learning subsystem **126** represents handshaking signals for coordination purposes, in addition to a data path for passing the filtered data to the learning subsystem **126**.

The representation subsystem **104** is a multiresolutional storage structure for organizing the system knowledge which integrates the information of experiences, i.e., the data **16** organized by the preprocessing subsystem **102**. Although the representation subsystem **104** may receive information on-line, the information may also be loaded or partially loaded off-line and updated either directly through the preprocessing subsystem **102** or via the learning subsystem **126**. The representation subsystem **104** extracts goal-independent data from goal-dependent data in the form of generalized rules, or components of the automata transition function and state output functions. For instance, standard algorithms and procedures for inverting experimental causalities into statements of rules may be used to extract goal-independent data. More sophisticated procedures can be applied, linked with algorithms and procedures of fusing the components of experiences into associative clusters. Such clusters form a function-oriented, goal-independent provisional model of the plant or process **14**.

The necessary components of a single level of a multiresolutional representation may be identified as: a goal which is an optimal step on a path to a global objective; a set of criteria used to determine a path to the assigned objective at the current resolutional level; a description of the system at the resolution appropriate for the selection of an optimal path with respect to the objectives of the level; and optimization algorithms for synthesizing the optimal strategy based on the previous three elements.

In order to obtain such a provisional model, a mechanism of consecutive multivariable clustering is applied, interpretable as generalization. The provisional model is also based upon unifying groups of higher resolution information into lower resolution units. However, instead of using experiences having similar preconditions, action and effect as inseparable units, similarities and components of the experiences are used, which leads to rules generation without explicit demonstration of partial dependencies demonstrated in the raw data **16**. Multiple goal-independent and time-independent rules are clustered into more general rule statements which produce a hierarchical system for storage and retrieval of rules.

Preferably, the representation subsystem **104** further uses the process of local interpolation, which introduces local errors in the computed costs of transitions in the graph. In order to establish that the errors which are introduced in this matter do not accumulate, several corrective measures may be taken. The simplest of these measures is to introduce a randomized shift to the discretization that is used for the graphs. The resulting structure is, therefore, organized as a random constellation of points of the state space rather than a conventional grid. In order to test the structure for a uniform distribution of errors, it is preferred to synthesize test trajectories and examine their estimated costs in conjunction with previously examined trajectories of operation.

The planning subsystem **106** searches the knowledge base generated by the preprocessing subsystem **102** and the representation subsystem **104** to select an optimal course of action (trajectory) and generates corresponding feed-forward control signals **114** which are sent to the arbitration subsystem **108** to carry out a predetermined assignment **124**. The assignment **124** for the system **100** includes a description of a task to be performed, as well as any additional information, such as performance criteria or special considerations. The knowledge base may be searched either on-line or off-line. The planning subsystem **106** requires input from the representation subsystem **104** and possibly, from the learning subsystem **126**. The output of the planning subsystem **106** comprises planned trajectories for variables and the feed-forward control signals **114**, as well as the local information **120** retained from its most recent search.

The following properties of tessellation are provided by the preprocessing and representation subsystems **102**, **104** which support the minimum-cost search for the planning/control trajectory. First, for systems with continuity, a recursive process of tessellation is assigned with a metric which generates a hierarchy with a virtue of addressability of all tessellata. The "all-through" addressability of the hierarchy is achievable using a decomposition strategy which allows for the property of having the state space address fully contained in the tessellatum name together with the level to which the state space belongs. Thus, the subsequent search cannot miss or misrecognize a tessellatum. Thus, the algorithm of reconsidering the hierarchy and adjusting the hierarchy is built into the system. Second, among all possible hierarchies which can be achieved by a space tessellation, there exists one hierarchy which delivers the most efficient (minimum cost) operation to the system to be controlled. Thus, the system is tuned in such a way as to have the

above-mentioned number of hierarchical levels and the algorithm of tuning up the hierarchies is built into the system. Finally, by associating the tessellata with their centers as the representatives lead to detrimental results; the excessive organization of the space creates biased motion (the well known idiosyncracies of the grids). Therefore, representing the tessellata not by their centers but rather, by randomized representatives heals the process of computation and motion in the system. Thus, the algorithm of randomization in assigning the representative for a tessellatum is also built into the system.

The present invention embeds the results of the examining operation under different regimes exercised by a lower level control subsystem. Each elementary transition which fits within the constraints of being a transition from one tessellatum to another, is assigned the values of state transition in the corresponding costs obtained from the on-line measurements (at the particular resolution determined by the level of tessellation). The value of error is assigned in such a way as not to exceed the "size of a tessellatum" accepted at a particular level of resolution. Thus, a device for embedding local dynamics is also built into the system.

Several methods of search for optimal trajectories may be applied to the graphs pertaining to a particular level of resolution. For instance, Dijkstra's algorithm (or its heuristic evolvement, A-star algorithm) may be used, as follows:

1. Initialize a searah tree of size equal to the number of nodes in the graph with the edge from the start node to itself, with a transition cost of zero, and mark it "open" for future examination;
2. Select the cheapest "open" node in the tree and mark it "closed" (investigated);
3. If the goal node is achieved, traverse pointers to retrieve the solution path;
4. Expand the current node, generating all of its successors (children) from the graph;
5. For each child which has not been previously expanded
    5.1 If the child is not in the "to" column of the tree, add the new edge, else
    5.2 If the existing cost to reach the child is higher than the newly determined cost, switch the source node and the cost on the existing edge;
6. Go to step (2).

In the presently preferred embodiment of the invention, the above algorithm is modified to incorporate suboptimal variations by implementing changes in step (5). The purpose for the modification to step (5) is to allow successive refinement of the results found at low resolution to provide increasingly precise approximations to the globally optimal strategy of operation. Thus, step (5) is preferably modified, as follows:

5. For each child
    5.1 If the number of occurrences of the child are less than the prearranged limit, add the new data to the tree and mark the new child "open";
    5.2 Else if the new child has a lower cost than all existing ways of getting to that child then
        5.2.1 If the same (to, from) pair exists in the tree replace it with the new data and mark the new child "open";
        5.2.2 Else if two identical (to, from) pairs exist in the tree replace the more expensive one with the new data and mark the new child "open";
        5.2.3 Else replace the most expensive (to, from) pair with the new data and mark the new child "open";
    5.3 Else
        5.3.1 If the same (to, from) pair exists in the tree and it is more expensive replace it with new data and mark the new child "open";
        5.3.2 Else if two identical (to, from) pairs exist in the tree replace the more expensive pair with the new data;
            5.3.2.1 If the replaced edge was more expensive, mark the new child "open";
            5.3.2.2 Else add the incremental cost of the more expensive substitution to edges built from the previous node, if it has already been expanded;
        5.3.3 Else if the new pair is less expensive replace the most expensive (to, from) pair with the new data and mark the new child "open".

An envelope surrounding the result at each level of resolution is determined using the following algorithm:

1. Initialize a table of size equal to the number of nodes in the search graph with the goal node and the cost of including in the optimal path to itself as C_{est} (x,x)=0 and C_{min} (x,a)=cost of the optimal path.
2. For each open node, k, in the table:
    2.1 Mark the node closed;
    2.2 Compute an estimate for the cost of reaching the goal from each predecessor node, k-1, using $C_{est}(x,k-1) = C_{est}(x,k)+c(k,k-1)$; Derive c(k,k-1) from the original graph and determine, from the search graph, the optimal cost C_{min} (k-1,a) of reaching k-1;
    2.3 If the existing value of C_{est} (x,k-1) is less than currently in the table, replace it, or if k-1 is not in the table, add it to the table and mark the node open;
3. Go to step 2.

The solutions are then refined in the new envelope by tessellating the space confined within the envelope at a finer level of resolution using the distributed store of data to fill in that volume only, thereby controlling the potentially explosive growth of complexity associated with conventional strategies of search for optimization. The width of the envelop is chosen based on known specifics of the problem. By providing search in the state space, analytical models are not required, and computations are both simplified and continually updated.

The prediction subsystem **110** generates suggested prediction control signals **116** for achieving optimal performance or operation of the plant **14**. Local information or data **120**, which is a byproduct of increasingly focussed search by the planning subsystem **106**, is transferred to the

prediction subsystem **110** as a set of increasingly precise investigations of successively tighter envelopes of a nominal trajectory. The prediction subsystem **110** operates on the principle that the best local course of action can be preplanned to multiple short horizons in real-time by investigating the local data or information **120** generated by the planning subsystem **106** in conjunction with the plant data **16** and the predetermined assignment **124**. Preferably, the prediction subsystem **110** is a predictive compensator.

The stabilization subsystem **112** is an optional element of the control system **100**. The stabilization system **112** generates a set of safety control signals **118** for assuring that the system operates within a predetermined range, i.e. a range of safe operation. For instance, the safety control signals **118** may comprise shut-off signals for turning off a piece of equipment or machinery which is operating outside of a predetermined safe range of operation, such as outside of a predetermined temperature range. Use of the stabilization subsystem **112** presumes the knowledge of some suboptimal but safe compensation to which the system **100** may revert in the event of unexpected or destabilizing behavior. Alternatively, the plant **14** may be equipped with stabilizing feedback before the synthesis of the optimizing control signals **114**.

The learning subsystem **126** correlates actual and expected performance data and continuously updates the preprocessing subsystem **102**, the representation subsystem **104** and the planning subsystem **106**. Preferably, a display **70** is coupled to the learning subsystem **126** which preferably combines graphical and textual information to allow a user to monitor operation of the system **100**.

The arbitration subsystem **108** receives the feed-forward control signals **114** and generates corresponding process control signals **28** for controlling the operation of the plant or process **14**. The arbitration subsystem **108** depends on the mode of operation of the overall system **100**. In its simplest configuration, the arbitration subsystem **108** superimposes the excitation or feed-forward control signals **114** suggested by the planning subsystem **106** on the control signals **116**, **118**, respectively, computed by the compensatory elements, i.e., the prediction subsystem **110** and the stabilization subsystem **112**. If, because of distributed decision making between the sources of actuation commands, prespecified limits such as hard bounds on inputs are violated, the arbitration subsystem **108** has the ability to modify the effective actuation accordingly. In case of unexpected behavior from the plant **14**, the arbitration subsystem **108** can be used to switch out individual components of the actuating signals **114** or to halt operation of the plant **14** completely. Thus, the arbitration subsystem **108** acts as a summer or switch to the plant **14**.

The control strategy of the present invention differs from the prior art (which tends to focus on analytical representations and parametric equations) by developing solutions (control signals) from the acquired data **16** at multiple resolution levels and sharing the control signals **114** in both an upward and downward fashion to satisfy the requirements of the system **100** as a whole, as well as system components.

The present invention eliminates the stage of mathematical abstraction and parameter identification and instead, uses collected measurements (the data **16**) to form a distributed multiresolutional knowledge representation which is operated upon by search algorithms. The present invention eliminates the need for an explicit choice of a mapping from empirical data to a particular model structure and the conventional equations are replaced by a distributed representation.

The operation of a power generating station provides opportunities to study the symbiotic relationships of dependent and independent process variables at multiple and discrete levels of resolution. The required megawatt (MW) output demand of a power plant is controlled by a unit load control system. The output demand value is used in a constraining function to control power plant subsystems (e.g., fuel/air, feed water, turbine valves, etc.) such that the required megawatt demand is met. The need for an integrated, multiresolutional view of complex systems emerges as a result of attempting to obtain optimal performance from the system as a whole when it may, without loss of generality, be viewed as a collection of interacting subsystems-each with its own parameters, constraints, optimization criteria, cycle times, and each itself composed of interacting subsystems.

A typical organizational approach of operating power plants at multiresolutional levels of information comprises operations designed and submitted for execution at succeeding resolutional levels on the basis of a priori information concerning safe and achievable targets. By not implementing a representation for the incorporation of feedback from lower levels, higher level decision makers are typically not in a position to set targets which are truly the "best" by objective criteria. Examples of such criteria for a power plant are: maximum efficiency of energy conversion; minimum emissions; maximum reliability of equipment and systems; and maximum flexibility of operation. In general, the list of potential performance criteria is large, but not as large as the number of controlled variables in the power plant, which number in the thousands. The organization of the variables into a hierarchy of significance (with respect to the most general aims of the process) is the first step in building a hierarchical representation of the system. This hierarchy of significance, while grounded in the basic physics of the process, is made at each level of the organizational hierarchy. As a result, although the fuel/air ratio has a very significant effect on unit performance, the scheduling of plant operations by the management level is typically driven by a monthly, or quarterly value of average efficiency, rather than variables of that nature.

Referring now to FIG. **4**, an electric grid **150** which supplies power to a region is managed by a power company **152**, with power generated by several utility companies **154**, **155**, **156**, each of which generally includes a plurality of power plants **158**–**164**, with each power plant **158**–**164** generally including a plurality of power generating units (not shown). The power company **152** is responsible for maintaining stable grid conditions through the use of base loaded, cycling and peaking power plants. Loading of the grid **150** is signaled by the frequency of the power supply by the utilities **154**–**156**. Generally, there is a band around the 60 Hz bus frequency within which the overall system must operate, and changes in this frequency indicate that the power generation capacity of the system must be adjusted. The individual utility companies must meet changes in capacity within an established time period.

The average cost of operation of a power plant under a utility company's control may be based on a parameter called lambda which indirectly reflects the cost of generation and is a function of both demand and production. The scheduling of individual plants is performed on the basis of a single averaged indicator.

Referring now to FIG. **5**, a schematic block diagram of a power generating unit **170** is shown. Generally, the power generating unit **170** comprises a source of fuel **172**, such as a natural gas line or a fuel storage facility, fuel flow controls,

such as a gas flow meter **174** and an oil flow meter **176** for regulating and measuring fuel flow. The power generating unit **170**, in addition to fuel resource elements, further comprises power generating elements, such as a boiler **178** for heating feed water **179**, a turbine **180** driven by the heated water or steam, and a generator **182** coupled to the turbine **180** for generating or producing output power **184**. The boiler **178** is also connected to an exhaust stack **186** for dissipating emissions, such as NOx and Sox. The power generating unit **170** has a predetermined range of economical operation. For instance, the unit **170** may produce power in a range from 90 MW–425 MW, with changes within the output power range varying in multiples of 40 MW. Thus, for example, an operating point change could be from 320 MW to 360 MW or from 120 MW to 400 MW.

Operating point changes are usually controlled based on previously established safety/economic criteria. For instance, operating point changes may be limited to 1% per minute (or 4 MW per minute). The operation is preprogrammed based on a knowledge of BTU/MW and a fuel heat content figure in BTU/fuel unit. Thus, the turbine **180** ramping and fuel increase is performed simultaneously (coordinated control), by following a stored curve for each. The BTU/MW is a steady-state figure. Within the power generating unit **170**, steam temperature should ideally be at 1000° F. for the first stage of the steam turbine **180** and the same at the outlet of the reheater for the secondary stages. However, at 20% load, the heat produced may not be adequate and the steam temperature at the reheater is allowed to go as low as 940° F. Similarly, the boiler **178** pressure falls from 2400 psi at full load to around 1050 psi at low load.

Feed-forward control (a reference curve or trajectory) is used for feed water **179**, drum pressure, steam temperature and air flow excess in addition to fuel. The fuel supply system **172** is operated with feed-forward and feedback control. First, volumetric consumption for steady-state production of a particular power output **184** is determined from a stored curve, expressed in terms of the rate at which fuel (both oil and gas) is burned to have a sustained supply of heat for the boiler **178**, as the boiler **178** supplies the correct steam conditions to the turbine **180**. A volumetric set point is provided to a loop that measures differential flow (consumption). Individual burners are capable of providing 150 MBTU with a 50% turn down. This means that the operating range of a burner is from MAX/2 to MAX (in gallons/min) and if "n" burners are in service, (n×MAX) down to (n×MAX/2) gallons may be forced out of the fuel supply nozzles. If more or less fuel is to be burned, additional burners must be brought into or taken out of service. In order to change the flow through the nozzles, the feedback system changes the differential pressure across the whole set of open nozzles until enough fuel is forced out of the nozzles. The amount of fuel to be burned is primarily set by the reference curve, however, the number of burners which can supply that volume is not unique. The operator in the control room is able to use his experience of usual demand and, if he expects a large increase in demand, to introduce several burners early (thereby lowering the flow/burner) in anticipation of an increase. The flow controller compensates for the number of burners by changing fuel pressure to contain fuel consumption at the feed-forward set point.

As previously discussed, many systems and processes possess symbiotic relationships of both dependent and independent data. In the power plant application, independence of variables is minimal. Unit load demand is determined by requirements of the regional electrical grid **150** and is

assigned based upon judgments of the power company **152** and the utility companies **154–156**. In the power plant generating control system in accordance with the present invention, three resolution levels of interest are identified on the basis of current industry practice, as follows: (1) the grid level at which a schedule of operation for each power generating unit is developed; (2) the unit level where the required power capacity should be developed in the most cost effective manner; and (3) the unit system level which has been broken down into entities such as fuel management system, feed water system, reheat control system, etc. Because optimization of fuel consumption directly translates to the global objective of improvement in unit operating efficiency, an intermediate unit level of resolution is analyzed and recommendations (e.g., control signals **114**) are made concerning the operation of a single unit in a manner which reduces costs, in terms of dollars per megawatt sold. As will be apparent to those of ordinary skill in the art, the described analysis and optimization techniques can be applied at each level of resolution, and therefore, extensions of this analysis can be made both up and down the hierarchy of concerns. Having a means of optimizing unit performance, it is possible to develop an optimal unit scheduling strategy (a higher level concern) using the same tools of analysis. Similarly, the optimal manner of controlling the unit systems is determined using the unit scheduling strategy as the goal to be achieved at the next level of resolution.

Initially, sufficient data must be collected to build a model of the process for efficiency analysis. In the electric power example, currently known data collection techniques may be used. For example, a Bailey Network distributed control system may be used to collect and maintain a history of the operating parameters for a power plant. Such a control system provides sufficient data for construction of a model. The variables initially selected for analysis in this example are based on parameters traditionally used by power companies to provide an overview of the unit condition. This results in the selection of a redundant set of variables for which data is collected by a plant control system. For instance, in the presently preferred embodiment, data is collected at one minute intervals over a predetermined period, such as a five day period. These variables are: (1) gross load; (2) auxiliary load; (3) main steam flow; (4) main steam pressure; (5) main steam temperature; (6) feed water flow; (7) final feed water temperature; (8) boiler drum pressure; (9) air heater output temperature; (10) turbine throttle pressure; (11) fuel temperature; (12) unit fuel gas flow; (13) unit fuel oil flow; (14) excess air; (15) SOx; and (16) NOx.

These parameters are culled based on their direct impact on the global objective of improved unit efficiency in the opportunity to control their value. This global objective is formulated for a level of resolution under consideration. For example, while SOx and NOx are important traditional parameters which characterize an emissions system, SOx and NOx do not directly impact on the unit efficiency (although emissions are monitored to ensure compliance with environmental regulations and to ensure that regulatory constraints are not a significant factor in the vicinity of the determined optimal operating trajectory). Also, while excess air flow does impact directly on the efficiency of the power generating unit **170**, air flow is slaved to fuel flow and provides little opportunity for control. Of course, some dependency relationships exist among the selected variables. Moreover, some variables are difficult to control or are directly related to other parameters. For instance, the tem-

perature and rate of flow of the feed water **179** are functions of the environment (e.g., water temperature varies with the season, for example from 85° in the summer to 35° in the winter), and are generally uncontrolled by a power plant.

The present invention determines a minimal set of variables which are used to carry out the desired optimization. The elimination of redundant variables allows optimization within a lower dimensional space, resulting in exponential reductions in complexity. If the performance of a power unit is not strongly correlated with certain variables, or if such variables can be independently controlled, then the examination of the effects of the variables can be deferred to the next level of resolution without significantly affecting the results of optimization.

The present invention accumulates selected transients from the acquired raw data **16**. For example, the generator gross power output **184** is identified as the output of the system. The input is the total heat supplied per unit timed by the combination of fuels. Controllable parameters having a direct impact on the efficiency of the unit **170** are turbine throttle pressure and main steam temperature. These variables were selected based first on assumptions concerning thermodynamic state variables and second, on the fact that the other variables examined may be treated as constraints which must simply be maintained within their normal operating range.

From specifications and actual measurements, the operational limits for each variable were selected as follows:

$80 \leqq x_1 \leqq 450$ (MW)

$950 \leqq x_1 \leqq 1020$ (° F.); and

$950 \leqq x_3 \leqq 2450$ (PSI).

Changes in the operating point of the unit **170** are mapped into a close subset of a three dimensional state space. The goal of optimization is stated as the determination of the trajectory from a particular power, temperature, and pressure to another, in a manner which minimizes fuel consumption per net MW during the transition along the complete trajectory from the beginning to the end.

Optimization is carried out using a multiresolutional representation built from the accumulated data **16**, rather than calculated from a mathematical model. Search algorithms allow the synthesis of trajectories at several levels of precision. The present invention utilizes empirically determined information (the data **16**) directly, and controls the growth of search complexity by recursively examining a smaller envelope in the state-space to determine the global optimum with increasing precision. The present invention performs the following steps:

(a) Tessellation or discretization;

(b) Determination of transition costs in the tessellated space;

(c) Search for the optimal trajectory at the current level of precision;

(d) Determining the envelope around the trajectory for using at the subsequent stage of refinement; and

(e) Successive refinement of the trajectory by utilizing a finer tessellation (or quantization level) within the envelope found at the previous stage.

Tessellation results in a partitioning of the state space into subsets identified by labels corresponding to their state-space location. The labels correspond to an indistinguishable "vicinity" in the state space and form an equivalences relation in conjunction with the indistinguishability measure. The smaller the tessellation size, the larger the set of labels, by an exponential factor corresponding to the dimensionality of the space. The valid transitions in the tessellated space are determined from the collected data **16**. The transitions are established by connecting randomly perturbed centers of the tessellata (vicinities or boxes) using evidence from the data **16**. Costs are assigned based on the accumulation of net costs per unit time over an interval required for the completion of each transition. Completion of a transition is defined as the point of closest approach to the state corresponding to a label, where distance is computed using the weighted 1-norm for the sake of simplicity. The input for each transition is approximated by a piecewise constant rate of combustion computed from the average of the rate at the start and end of every transition.

Referring now to FIG. **6A**, a plot of the gross power output **184** developed during a **30** hour period against time generated by the preprocessing subsystem **102** is shown. FIG. **6A** shows the typical behavior required from a power plant, such as one of the power plants **158–164**. In FIG. **6B**, the plot of power output **184** against time (FIG. **6A**) is overlaid on a corresponding generating cost in terms of BTU/MW. The cost per net MW is computed by sliding an adjustable width averaging window along a time axis and combining fuel BTU expenditure figures for both oil and gas on the basis of measured caloric value. FIG. **6B** illustrates that the per unit cost of production of power is significantly higher for power output below a level of about 160 MW than for power above such level. Cost peaks are associated with all increases in power output. The goal of optimization is to characterize the manner of increasing power output which minimizes the cost per net MW. From FIG. **6B**, it is apparent that a potential source of savings is associated with transients during load shifting, and especially during ramping. Thus, the system focuses on transient behavior and characterizes the manner of increasing output which employs the least energy. FIG. **6C** shows the accumulation of selected transients from the raw data **16**. Such selection of transients may be conducted manually by an operator or automatically, using pattern recognition algorithms. The selected transients are preferably combined and presented to the representation subsystem **104**.

The process of incorporating the data **16** into a consistent representation is controlled by the representation subsystem **104**. The representation subsystem **104** abstracts the data **16** into a multiresolutional representation by consecutive bottom-up multivariable clustering (generalization). Referring now to FIGS. **7A–7C**, in the case of the power plant example, the state-space has only three dimensions and is given a tangible graphic interpretation. In FIG. **7A**, the entire space of practical operation (the work space) for the power unit **170** is shown. The axes are power **200**, pressure **202** and temperature **204**. The thermodynamic variables of pressure and temperature provide sufficient indicators of process dynamics to generate a first approximation. Pressure is measured at the turbine throttle and temperature at a preceding measurement stage. Individual points **206** within the space represent a label or marker to a distinct range of operating points. A range of operating points (per marker) is associated with the respective marker, using the measure of indistinguishability at a particular resolution level. For the first resolution level (FIG. **7A**), each point represents a range of power output of about 35 MW, 1350 psi, and 6° F. The points actually form a cloud which is depicted as a grid in order to simplify the presentation of the results. The lines or edges **208** connecting the points **206** in the grid mark the existence of stored information about ways of moving from one operating point **206** to another operating point **206** at the level of resolution (and their costs). This information is

stored with the limited precision of this rough level of resolution. The purpose of this information is to delineate a region of the space on which to focus while examining a particular range in the power output level **184**. FIG. **7**A illustrates only a limited amount of information since it is drawn from only a few days of operation. However, it will be apparent to those of ordinary skill in the art that much greater amounts of information can be handled by the present invention.

The acquisition and storage of the data **16** may be performed either automatically and/or manually and also may be performed either off-line and/or on-line. An important feature of the present invention is the ability to automatically integrate information while monitoring a process on-line. The automation of the preprocessing subsystem **104** and the structure of the constructed model allows the updating of information whenever changes in system behavior become apparent. FIGS. **7**B and **7**C show subsets of the space at increasing resolution. In general, having collected huge amounts of data, it is not computationally feasible to study the whole space when looking for particular optimal paths. Accordingly, the preprocessing subsystem **102** preferably includes an algorithm for pruning the space before proceeding at higher resolution levels constructing the envelope for the subsequent search). In the general case, the number of variables studied as a group is often larger than three. The functioning of the present invention is not affected by this fact, however, the convenient graphical symbolism is lost. Of course, groups of three or less variables may still be viewed at the same time. The purpose of FIGS. **7**A–**7**C is only to provide a clear and tangible interpretation of the approach used by the present invention. Between the first and third levels of resolution (FIGS. **7**A and **7**C), the tessellation sizes are halved between levels of resolution, such that the spatial density of the tessella increases four times between FIGS. **7**A and **7**C.

The purpose of constructing the multiresolutional representation, in general, is to allow the efficient use of search at the initial level for the optimal strategy for making the operating point low resolution transitions required during normal operation of the power generating unit **170**, followed by the selection of a subset of the representation at the next level for more refined search. If this strategy is not implemented, the number of points which must be searched for an adequately precise solution is prohibitive. The planning subsystem **106** performs a search at several levels of resolution before arriving at a solution (optimal trajectory) of appropriate accuracy. A variation on Dijkstra's search algorithm, as previously discussed, is applied to the distributed graph representation (FIGS. **7**A–**7**C) to determine optimal strategies of control.

FIG. **8**A shows the results of searching for a single solution trajectory **210** at the first level of resolution and FIG. **8**B shows the results **212** for searching for a single solution trajectory at the third level of resolution. The objective of the analysis is to find the most fuel efficient way of changing the operating point of the unit **170** from that corresponding to an output of 90 MW to one corresponding to an output of 400 MW. The throttle pressure change corresponding to a change in the power output **184** is from 1000 psi to 2400 psi and the temperature change is from 990° F. to 1000° F. The variation on the standard search algorithm is designed to allow the examination of suboptimal paths, which results in the ability to determine a locus of paths whose costs differ from an optimum (at the current level) by an amount that is less than an assigned threshold. This locus is used to select an envelope around the optimum which is expected to yield a more precise result at the next level of resolution. The refinement is continued until an acceptable degree of precision is attained. The total number of levels in a multiresolutional hierarchy is determined by the overall space of interest at the lowest level of resolution, by the final accuracy at the highest level of resolution, and by the condition of minimizing computational complexity. The search results at the third level of resolution are shown in FIG. **8**B. FIG. **8**B is a three-dimensional plot which superimposes the optimal state trajectory for an output transition from 90 MW–400 MW on the data **16**.

FIGS. **9**A and **9**B are two-dimensional plots (i.e. a decomposition of FIG. **8**B) of the power **200** against the pressure **202**, and the temperature **204**, respectively. FIGS. **9**A and **9**B are shown in order to provide a more intuitive feeling for the suggested strategy. FIGS. **9**A and **9**B show the optimal pressure-power plot and temperature-power plot, respectively, for a cycle of output change from 90 MW–400 MW and back down to 90 MW and illustrates how parameters and variables should move to maximize unit **170** efficiency. The approximate cost of the suggested path during ramp-up is about 110 MBTU-hr. From FIG. **9**A and FIG. **9**B, it is clear that the suggested strategy allows (acceptable) fluctuations in steam temperature, and imposes a linear relationship between pressure **202** and power **200** in all but the initial stages of an increase in the power output **184**.

The envelope creation algorithm, which is also used for selecting a subset of the space for further refinement during transitions from lower to higher resolution, provides data on other strategies attempted during the same increase of output. According to this information, the process of ramping the power generating unit **170** should be more uniform and closer to the optimal curves suggested. Accordingly, the control signals **114** generated by the planning subsystem **106** attempt to control the unit **170** in order to make the ramping up process more uniform. Additional data may be used to make this analysis more comprehensive and accurate (with additional levels of representation which are inappropriate without the corresponding data).

FIG. **10** shows the suggested heat input **214** required (in terms of BTU) for following the optimal trajectories determined by the control system. The heat input **214** is shown in terms of BTU and is plotted against power output. Using known fuel cost information, such as the cost of oil or gas, specific oil and/or gas flow rates may be calculated to determine the efficiency of the unit **170**.

From the foregoing description, it can be seen that the present invention comprises a multiresolutional decision support system for multiresolutional analysis of data from diverse sources or subsystems to formulate plans or rules of system operation which enhance system performance. The foregoing describes the application of multiresolutional analysis techniques to an operating power plant. However, the principles disclosed herein, e.g., data structuring and organizing, selective parameter lumping and search-based optimization methods can be applied to any system or process possessing the attribute of semiotic variables which result in a multiresolutional array of information. That is, other process systems, such as those found in fabricating plants, chemical facilities, refineries, and the like, can benefit in similar fashion to the analysis and modeling techniques described herein. The present invention is also useful for a number of other systems and activities which have not been previously recognized as possessing semiotic characteristics and which cannot be adequately modeled

mathematically, such as assembly, packaging and warehousing facilities and mission definition and analysis. It will be recognized by those skilled in the art that changes may be made to the above-described invention without departing from the broad inventive concepts thereof. It is understood, therefore, that this invention is not limited to the particular embodiment disclosed, but is intended to cover any modifications which are within the scope and spirit of the invention as defined by the appended claims.

What is claimed is:

1. In a power plant, a planning and control system for determining output trajectories of a predefined output behavior and input control commands for achieving the predefined output behavior, the system comprising:

(a) means for acquiring data related to at least one process variable of the power plant;

(b) means for organizing the acquired data and generating at least one state input-output string therefrom;

(c) means for fusing the strings into associative clusters, the clusters forming a function oriented, goal-independent provisional model of the power plant;

(d) means for generating a multiresolutional data structure based upon the associative data clusters for transforming the provisional model into at least one level of resolution incorporating embedded local dynamics of the power plant;

(e) means for searching the multiresolutional data structure at varying levels of resolution to determine an optimal trajectory of operation for achieving the predefined output behavior; and

(f) means for generating control signals for controlling at least one operational variable of the power plant such that the power plant operates according to the optimal trajectory.

2. The system of claim 1 wherein the data is acquired on-line and the search is conducted off-line.

3. In a power plant, a method for determining output trajectories of a predefined output behavior and input control commands for achieving the predefined output behavior, the method comprising the steps of:

(a) acquiring data related to at least one process variable of the power plant;

(b) organizing the acquired data and generating at least one state input-output string therefrom;

(c) fusing the strings into associative clusters, the clusters forming a function oriented, goal-independent provisional model of the power plant;

(d) generating a multiresolutional data structure based upon the associative data clusters for transforming the provisional model into at least one level of resolution incorporating embedded local dynamics of the power plant;

(e) recursively searching the multiresolutional data structure at varying levels of resolution to determine an optimal trajectory of operation for achieving the predefined output behavior; and

(f) generating control signals for controlling at least one operational variable of the power plant such that the power plant operates according to the optimal trajectory.

4. The method of claim 3 wherein step (e) further comprises synthesizing optimal trajectories of process operation using a search applied to the multiresolutional data structure at successively higher levels of resolution with subsequent propagation of the search results to successively lower levels of resolution.

5. The method of claim 3 wherein the data is acquired on-line and the search is conducted off-line.

* * * * *

Evidence Appendix 2

Johnston

US006826541B1

(12) **United States Patent**
Johnston et al.

(10) **Patent No.: US 6,826,541 B1**
(45) **Date of Patent: Nov. 30, 2004**

(54) **METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR FACILITATING USER CHOICES AMONG COMPLEX ALTERNATIVES USING CONJOINT ANALYSIS**

(75) Inventors: **Jeffrey M. Johnston**, Durham, NC (US); **Adam B. Richman**, Charlotte, NC (US); **David L. Sheeks, III**, Durham, NC (US); **Richard R. Shopmyer**, Durham, NC (US)

(73) Assignee: **Decision Innovations, Inc.**, Durham, NC (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 527 days.

(21) Appl. No.: **09/704,349**

(22) Filed: **Nov. 1, 2000**

(51) **Int. Cl.**[7] **G06F 17/35**

(52) **U.S. Cl.** **705/10**; 705/1; 705/11; 705/35; 705/36; 705/2; 705/4; 702/81; 702/82; 702/83; 702/84; 702/179; 702/180; 702/181; 702/182; 702/183; 702/184; 702/185; 702/186; 702/188

(58) **Field of Search** 705/10, 11, 1, 705/35, 36, 2, 4; 702/81–84, 179–181, 182–186, 188

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,495,412 A	*	2/1996	Thiessen	705/1
5,717,865 A	*	2/1998	Stratmann	705/10
5,742,776 A	*	4/1998	Toda	705/1
5,799,311 A	*	8/1998	Agrawal et al.	707/102
5,832,447 A		11/1998	Ricker et al.	
5,926,794 A	*	7/1999	Fethe	705/11
6,064,984 A	*	5/2000	Ferguson et al.	705/36
6,249,768 B1	*	6/2001	Tulskie et al.	705/7
6,397,212 B1		5/2002	Biffar	
2001/0010041 A1		7/2001	Harshaw	
2002/0004739 A1		1/2002	Elmer et al.	
2002/0087388 A1		7/2002	Keil et al.	

OTHER PUBLICATIONS

O'Hara, Colleen, "Comparison shop for health plans online", Federal Computer Week, Mar. 22, 1999, vol. 13, No. 7, 2 pages, retrieved from: Proquest.*
Appleby, Chuck, "Software for hard choices", Hospitals & Health Networks, Jun. 5, 1998, vol. 72, No. 11, 2 pages, retrieved from: Dialog, file 15.*
"Partnerships '98, Showcase and Games", www.odphp.osophs.dhhs.gov/confrnce/partnr98, Apr. 28, 1998, 1 page, retrieved from: google.com.*
"California Consumer Health Scope", www.healthscope.org, Feb. 3, 1998, 3 pages, retrieved from: archive.org and google.com.*
"Decision Innovations signs agreement with U.S. Office of Personnel Management", PR Newswire, Nov. 12, 1998, 2 pages, retrieved from: Dialog, file 813.*
"Software by Decision Innovations Inc. Helps Employees Choose Health plans", PR Newswire, Dec. 9, 1997, 3 pages, retrieved from: Dialog, file 813.*

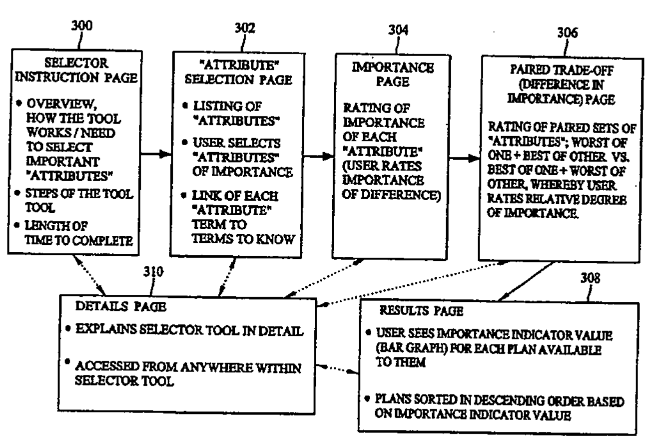
(List continued on next page.)

Primary Examiner—Tariq R. Hafiz
Assistant Examiner—Beth Van Doren
(74) *Attorney, Agent, or Firm*—Jenkins, Wilson & Taylor, P.A.

(57) **ABSTRACT**

Methods, systems, and computer program products for facilitating user choices among complex alternatives utilize conjoint analysis to simplify choices to be made by the user. A selector tool presents a user with a first and second series of choices relating to attributes of products or services available to the user. A utilities calculation engine calculates the relative utility of each of the products or services to the user and presents output to the user, which indicates the relative utility of each of the products or services. The user can then select the product or service that has the highest utility value for the user based on the calculated relative utility values.

86 Claims, 9 Drawing Sheets



OTHER PUBLICATIONS

Screenshots of Decision Innovations, www.decision-innovations.com, Dec. 6, 1998–Feb. 8, 1999, 10 pages, retrieved from: google.com and archive.org.*

"OPM makes life easier for people pondering health care open season choices", OPM News Release, Nov. 9, 1998, 2 pages, retrieved from: google.com.*

"News Briefs: Tool Available for Selecting a Health Plan", NIH Record, Nov. 3, 1998, 6 pages, retrieved from: google.com.*

"Federal Employees Health Benefits (FEHB)", www.opm.gov/insure, Feb. 2, 1999, 3 pages, retrieved from: archive.org and google.com.*

"Decision Innovations Signs Agreement with National Institute of Health", PR Newswire, Oct. 30, 1998, 2 pages, retrieved from: Proquest.*

Bard, Mark, "Purchasing Health Care and Value", Jan. 1998, vol. 77, Issue 1, pp. 1–5, retrieved from: Proquest Direct.*

"FEHB Program Carrier Letter", US Office of Personnel Management, May 31, 2000, pp. 1–2, retrieved from: Google.com.*

"FEDmanager", www.fedmanager.com, Nov. 16, 1999, pp. 1–3, retrieved from: Google.com.*

ANONYMOUS, www.asparity.com, web page printout (Feb. 6, 2002).

ANONYMOUS, "The Power of Real-Time Collaborative Customer Conversations," p. 1–2, (Jan. 31, 2002).

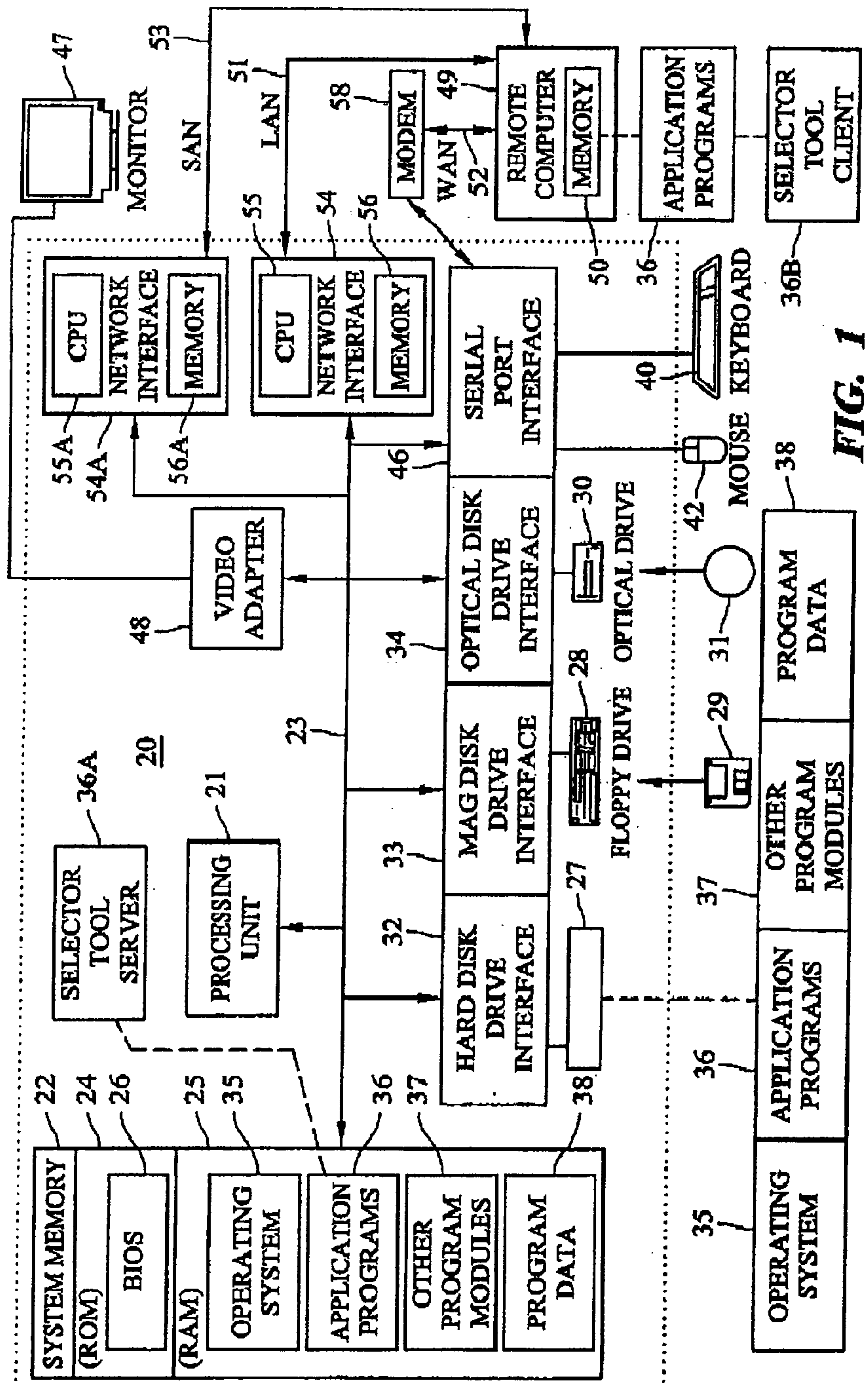
ANONYMOUS, "Our Customers Benefit From Powerful Partnerships," p. 1, (Jan. 31, 2002).

ANONYMOUS, "Online Insight and Top 10 National Homebuilder Break New Ground Launching Online Home Finder Solution," p. 1–3, (Jan. 31, 2002).

ANONYMOUS, www.beazer.com, web page printout, (Jan. 28, 2002).

ANONYMOUS, www4.activebuyersguide.com, web page printout (Jan. 24, 2002).

* cited by examiner



SYSTEM MEMORY
(ROM)
BIOS
(RAM)
OPERATING SYSTEM
APPLICATION PROGRAMS
OTHER PROGRAM MODULES
PROGRAM DATA
SELECTOR TOOL SERVER
PROCESSING UNIT
HARD DISK DRIVE INTERFACE
MAG DISK DRIVE INTERFACE
OPTICAL DISK DRIVE INTERFACE
SERIAL PORT INTERFACE
VIDEO ADAPTER
CPU
NETWORK INTERFACE
MEMORY
MONITOR
SAN
LAN
MODEM
WAN
REMOTE COMPUTER
MEMORY
APPLICATION PROGRAMS
SELECTOR TOOL CLIENT
FLOPPY DRIVE
OPTICAL DRIVE
MOUSE
KEYBOARD
OPERATING SYSTEM
APPLICATION PROGRAMS
OTHER PROGRAM MODULES
PROGRAM DATA
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
36A
36B
37
38
40
42
46
47
48
49
50
51
52
53
54
54A
55
55A
56
56A
58

FIG. 1

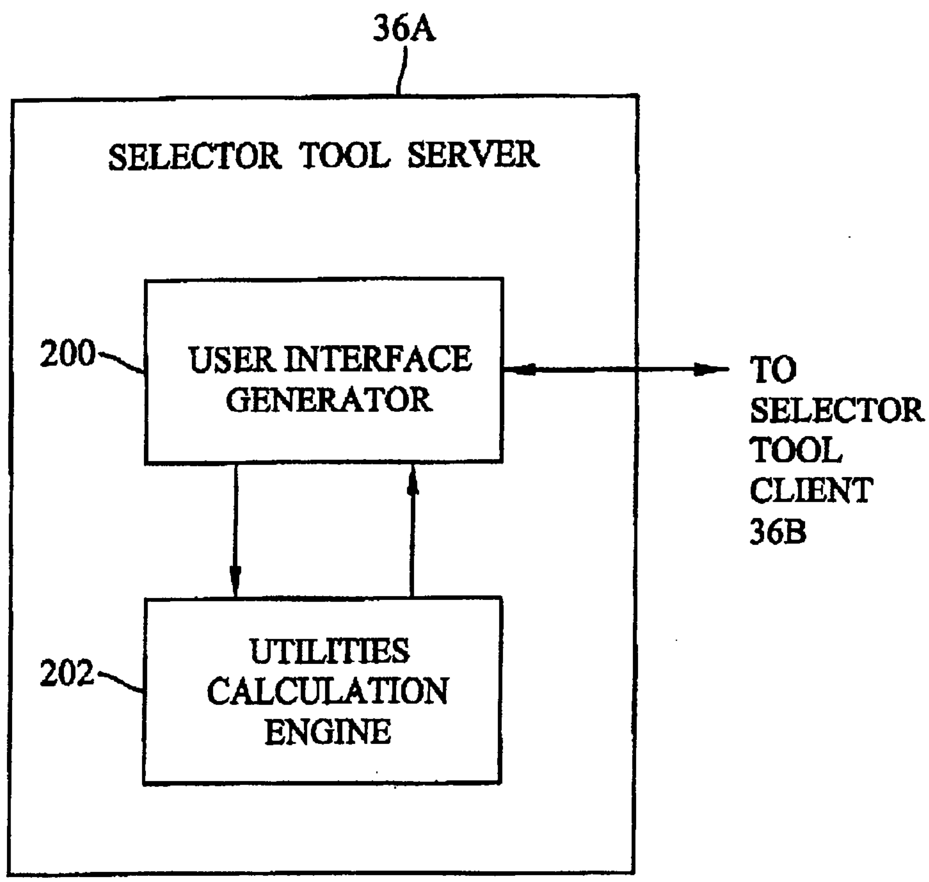

36A
SELECTOR TOOL SERVER
200
USER INTERFACE GENERATOR
202
UTILITIES CALCULATION ENGINE
TO SELECTOR TOOL CLIENT 36B

FIG. 2

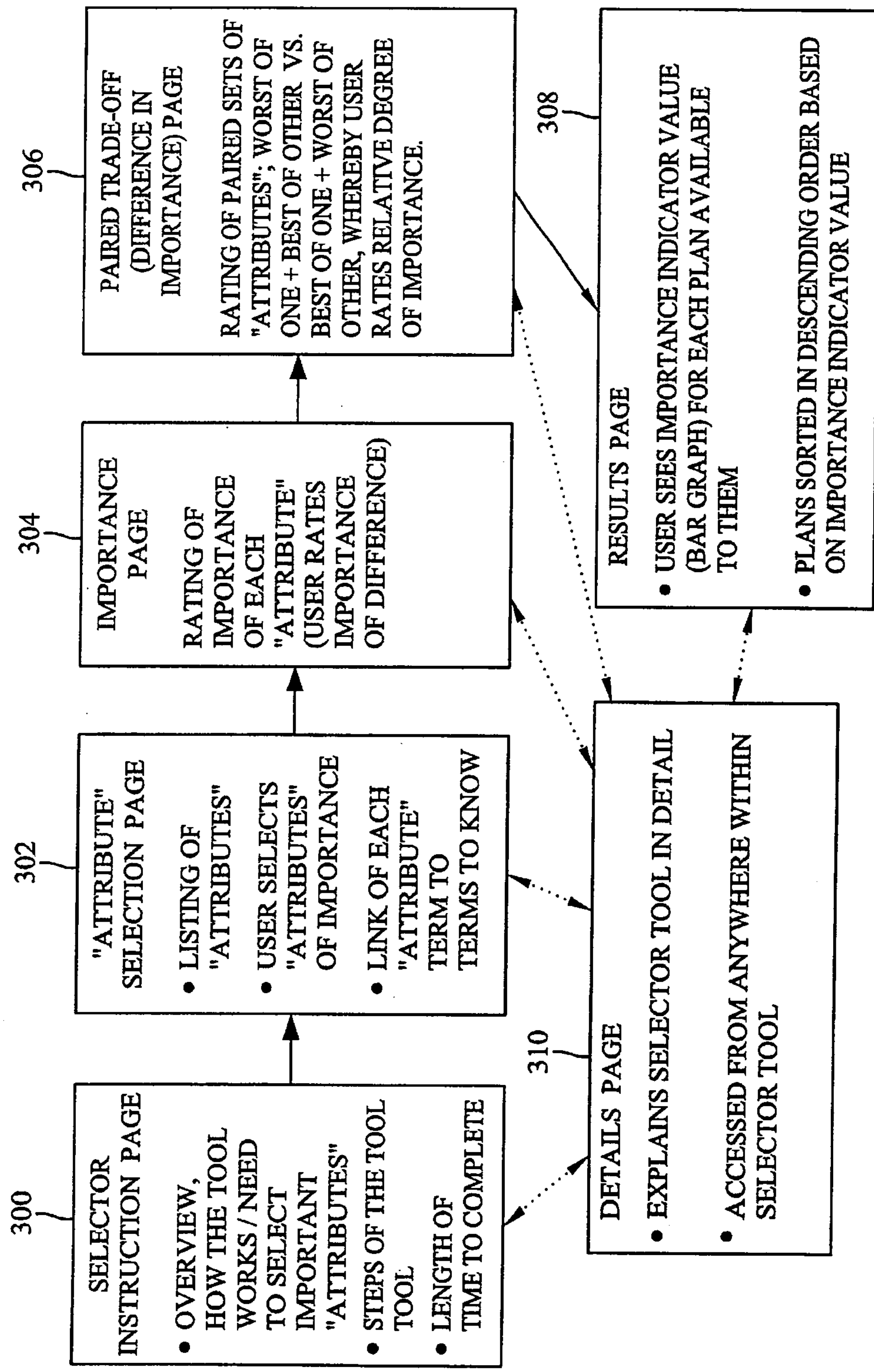

300
SELECTOR INSTRUCTION PAGE
• OVERVIEW, HOW THE TOOL WORKS / NEED TO SELECT IMPORTANT "ATTRIBUTES"
• STEPS OF THE TOOL TOOL
• LENGTH OF TIME TO COMPLETE
302
"ATTRIBUTE" SELECTION PAGE
• LISTING OF "ATTRIBUTES"
• USER SELECTS "ATTRIBUTES" OF IMPORTANCE
• LINK OF EACH "ATTRIBUTE" TERM TO TERMS TO KNOW
304
IMPORTANCE PAGE
RATING OF IMPORTANCE OF EACH "ATTRIBUTE" (USER RATES IMPORTANCE OF DIFFERENCE)
306
PAIRED TRADE-OFF (DIFFERENCE IN IMPORTANCE) PAGE
RATING OF PAIRED SETS OF "ATTRIBUTES"; WORST OF ONE + BEST OF OTHER VS. BEST OF ONE + WORST OF OTHER, WHEREBY USER RATES RELATIVE DEGREE OF IMPORTANCE.
308
RESULTS PAGE
• USER SEES IMPORTANCE INDICATOR VALUE (BAR GRAPH) FOR EACH PLAN AVAILABLE TO THEM
• PLANS SORTED IN DESCENDING ORDER BASED ON IMPORTANCE INDICATOR VALUE
310
DETAILS PAGE
• EXPLAINS SELECTOR TOOL IN DETAIL
• ACCESSED FROM ANYWHERE WITHIN SELECTOR TOOL

FIG. 3

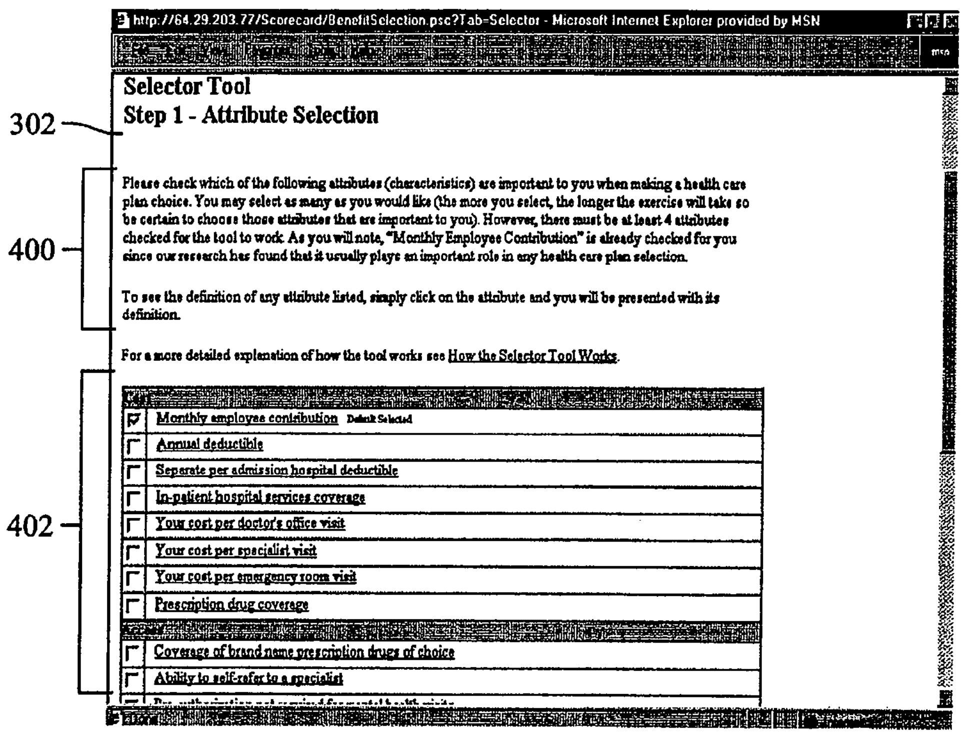

http://64.29.203.77/Scorecard/BenefitSelection.psc?Tab=Selector - Microsoft Internet Explorer provided by MSN
Selector Tool
Step 1 - Attribute Selection
302
Please check which of the following attributes (characteristics) are important to you when making a health care plan choice. You may select as many as you would like (the more you select, the longer the exercise will take so be certain to choose those attributes that are important to you). However, there must be at least 4 attributes checked for the tool to work. As you will note, "Monthly Employee Contribution" is already checked for you since our research has found that it usually plays an important role in any health care plan selection.
To see the definition of any attribute listed, simply click on the attribute and you will be presented with its definition.
400
For a more detailed explanation of how the tool works see How the Selector Tool Works.
Monthly employee contribution Default Selected
Annual deductible
Separate per admission hospital deductible
In-patient hospital services coverage
Your cost per doctor's office visit
Your cost per specialist visit
Your cost per emergency room visit
Prescription drug coverage
Coverage of brand name prescription drugs of choice
Ability to self-refer to a specialist
402

FIG. 4

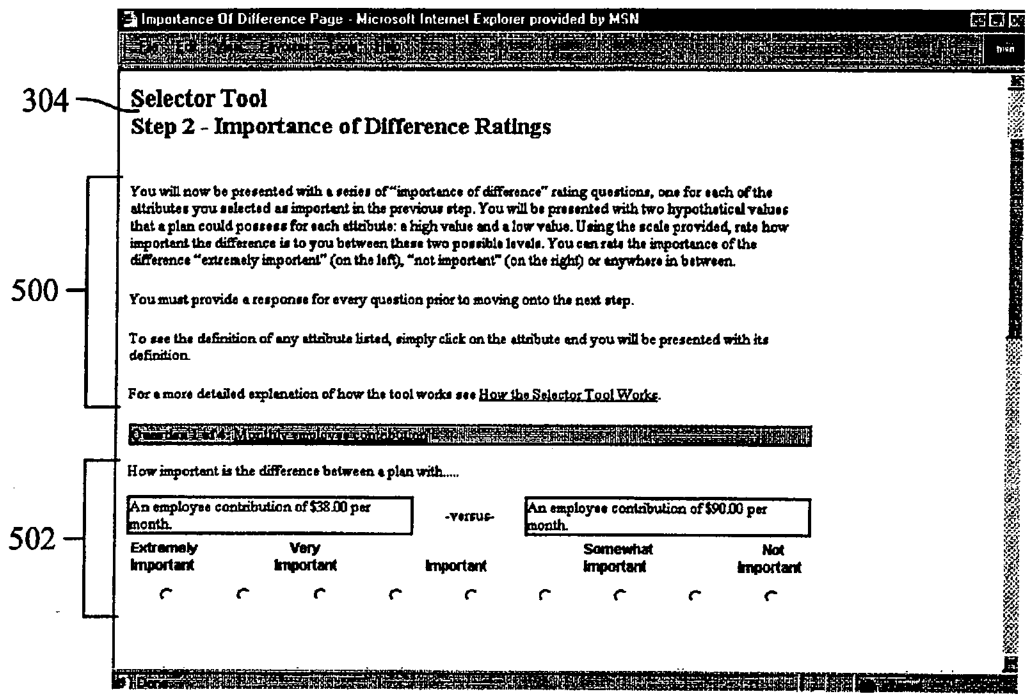
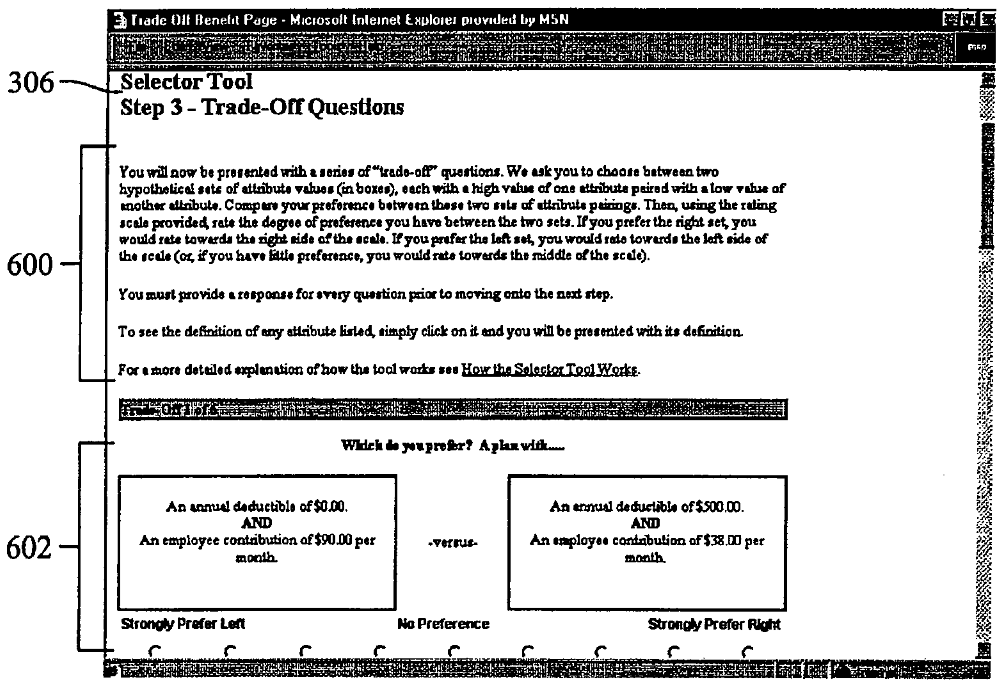

Importance Of Difference Page - Microsoft Internet Explorer provided by MSN
304
Selector Tool
Step 2 - Importance of Difference Ratings
500
You will now be presented with a series of "importance of difference" rating questions, one for each of the attributes you selected as important in the previous step. You will be presented with two hypothetical values that a plan could possess for each attribute: a high value and a low value. Using the scale provided, rate how important the difference is to you between these two possible levels. You can rate the importance of the difference "extremely important" (on the left), "not important" (on the right) or anywhere in between.
You must provide a response for every question prior to moving onto the next step.
To see the definition of any attribute listed, simply click on the attribute and you will be presented with its definition.
For a more detailed explanation of how the tool works see How the Selector Tool Works.
502
How important is the difference between a plan with.....
An employee contribution of $38.00 per month.
-versus-
An employee contribution of $90.00 per month.
Extremely Important
Very Important
Important
Somewhat Important
Not Important

FIG. 5



Trade Off Benefit Page - Microsoft Internet Explorer provided by MSN
306
Selector Tool
Step 3 - Trade-Off Questions
600
You will now be presented with a series of "trade-off" questions. We ask you to choose between two hypothetical sets of attribute values (in boxes), each with a high value of one attribute paired with a low value of another attribute. Compare your preference between these two sets of attribute pairings. Then, using the rating scale provided, rate the degree of preference you have between the two sets. If you prefer the right set, you would rate towards the right side of the scale. If you prefer the left set, you would rate towards the left side of the scale (or, if you have little preference, you would rate towards the middle of the scale).
You must provide a response for every question prior to moving onto the next step.
To see the definition of any attribute listed, simply click on it and you will be presented with its definition.
For a more detailed explanation of how the tool works see How the Selector Tool Works.
602
Which do you prefer? A plan with.....
An annual deductible of $0.00.
AND
An employee contribution of $90.00 per month.
-versus-
An annual deductible of $500.00.
AND
An employee contribution of $38.00 per month.
Strongly Prefer Left
No Preference
Strongly Prefer Right

FIG. 6

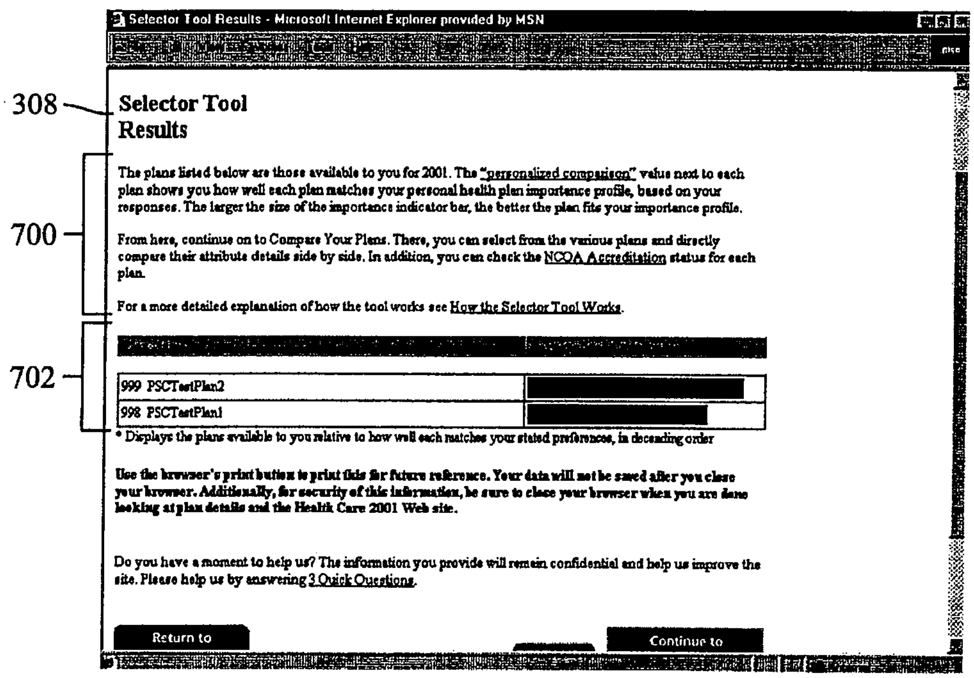

Selector Tool Results - Microsoft Internet Explorer provided by MSN
308
Selector Tool
Results
700
The plans listed below are those available to you for 2001. The "personalized comparison" value next to each plan shows you how well each plan matches your personal health plan importance profile, based on your responses. The larger the size of the importance indicator bar, the better the plan fits your importance profile.
From here, continue on to Compare Your Plans. There, you can select from the various plans and directly compare their attribute details side by side. In addition, you can check the NCQA Accreditation status for each plan.
For a more detailed explanation of how the tool works see How the Selector Tool Works.
702
999 PSCTestPlan2
998 PSCTestPlan1
* Displays the plans available to you relative to how well each matches your stated preferences, in decending order
Use the browser's print button to print this for future reference. Your data will not be saved after you close your browser. Additionally, for security of this information, be sure to close your browser when you are done looking at plan details and the Health Care 2001 Web site.
Do you have a moment to help us? The information you provide will remain confidential and help us improve the site. Please help us by answering 3 Quick Questions.
Return to
Continue to

FIG. 7

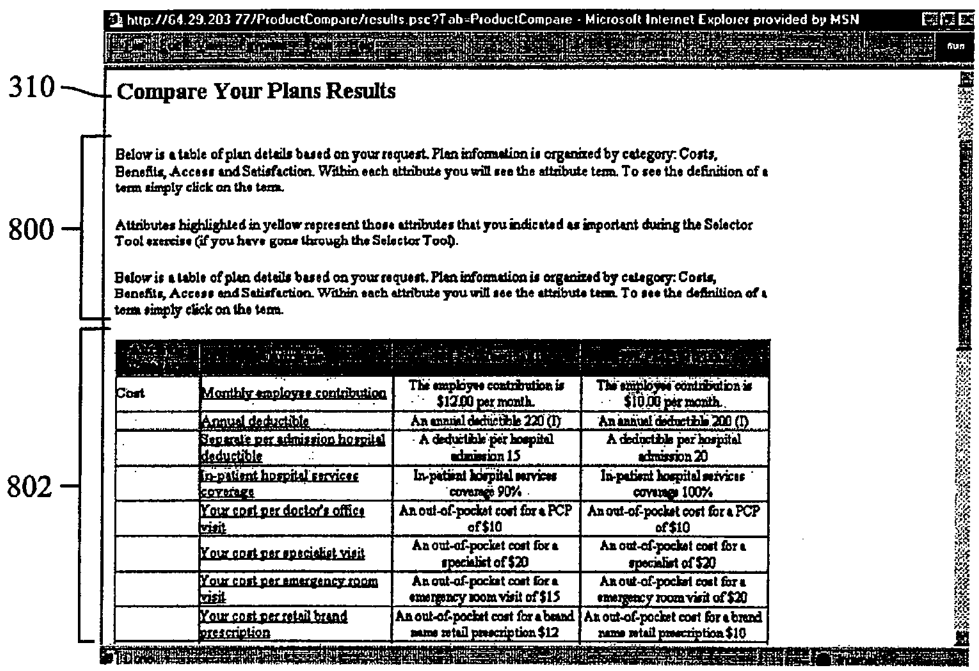

http://64.29.203 77/ProductCompare/results.psc?Tab=ProductCompare - Microsoft Internet Explorer provided by MSN
310
Compare Your Plans Results
800
Below is a table of plan details based on your request. Plan information is organized by category: Costs, Benefits, Access and Satisfaction. Within each attribute you will see the attribute term. To see the definition of a term simply click on the term.
Attributes highlighted in yellow represent those attributes that you indicated as important during the Selector Tool exercise (if you have gone through the Selector Tool).
Below is a table of plan details based on your request. Plan information is organized by category: Costs, Benefits, Access and Satisfaction. Within each attribute you will see the attribute term. To see the definition of a term simply click on the term.
802
Cost
Monthly employee contribution
The employee contribution is $12.00 per month.
The employee contribution is $10.00 per month.
Annual deductible
An annual deductible 220 (I)
An annual deductible 200 (I)
Separate per admission hospital deductible
A deductible per hospital admission 15
A deductible per hospital admission 20
In-patient hospital services coverage
In-patient hospital services coverage 90%
In-patient hospital services coverage 100%
Your cost per doctor's office visit
An out-of-pocket cost for a PCP of $10
An out-of-pocket cost for a PCP of $10
Your cost per specialist visit
An out-of-pocket cost for a specialist of $20
An out-of-pocket cost for a specialist of $20
Your cost per emergency room visit
An out-of-pocket cost for a emergency room visit of $15
An out-of-pocket cost for a emergency room visit of $20
Your cost per retail brand prescription
An out-of-pocket cost for a brand name retail prescription $12
An out-of-pocket cost for a brand name retail prescription $10

FIG. 8

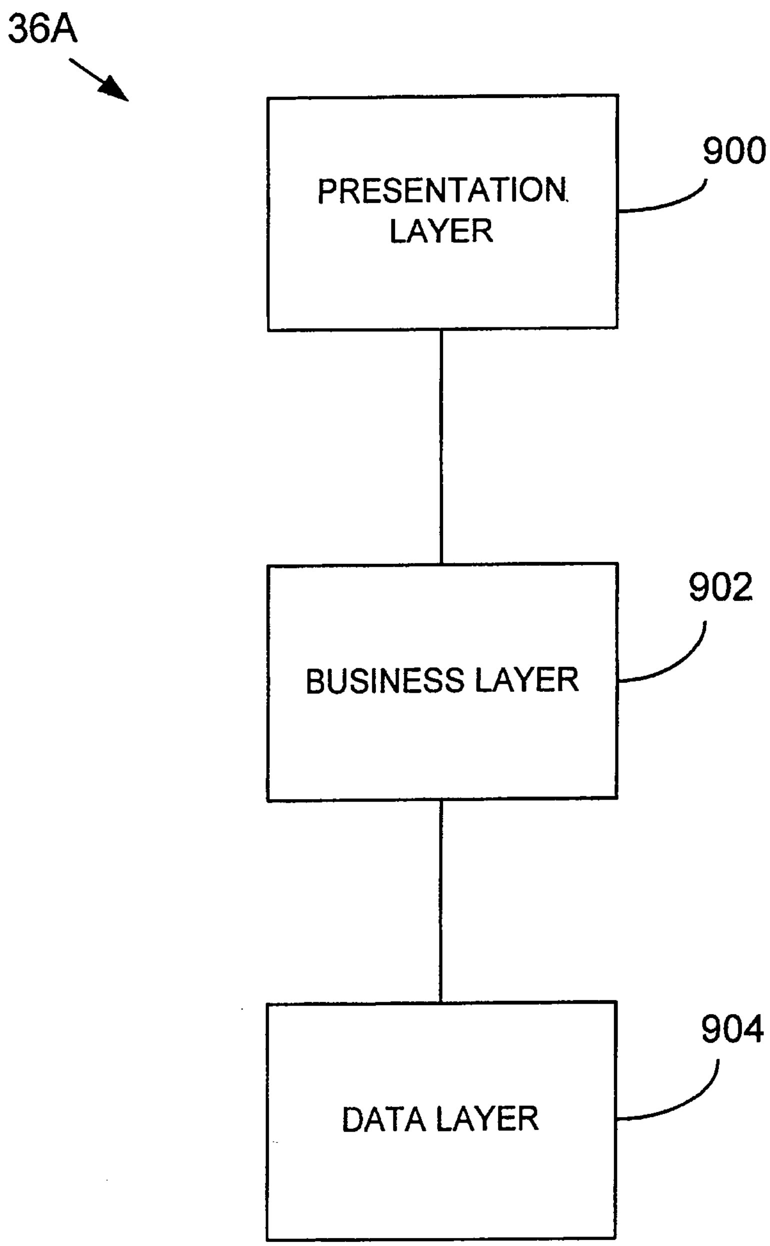

36A
PRESENTATION LAYER
900
BUSINESS LAYER
902
DATA LAYER
904

FIG. 9

METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR FACILITATING USER CHOICES AMONG COMPLEX ALTERNATIVES USING CONJOINT ANALYSIS

TECHNICAL FIELD

The present invention relates to using a proven social science statistical technique called conjoint analysis to facilitate choices among complex alternatives. More particularly, the present invention relates to methods, systems and computer program products for facilitating individual user choices among complex alternatives using a unique adaptation of conjoint analysis.

BACKGROUND ART

As a research methodology, conjoint analysis has been in use in the academic and commercial research community for many years (since the mid-1970's), and has been commonly used for marketing research purposes to assess consumer preferences among competing products or services.

Generally, conjoint analysis is a tool that researchers use to estimate the relative importance of the attributes that comprise the "alternatives" in the "choice set" and how much utility each "setting" of each "attribute" has for individuals. Results are often used to simulate the effect on market share that various changes in the "attribute settings" have and thus to fine tune "alternatives" (e.g. identify the optimal price for a product) and to forecast market share. While many forms of conjoint analysis exist, there are two general defining properties of any conjoint process: 1) each at some point gathers data from individuals by asking each individual to consider (the "con" in conjoint) two or more variables simultaneously or jointly (the "joint" in conjoint) and 2) each uses the gathered data (responses) to estimate how much utility or value each "attribute setting." Typically, conjoint data is gathered from a sample of users and then analyzed with no flow of information back to the user. Thus, there exists a long-felt need for applications that use conjoint analysis to facilitate individual user choices among complex decisions by providing conjoint analysis results back to the user.

For example, this need is particularly acute in the area of employer-sponsored health plans. Many large- and medium-sized employers offer a number of health plan options for employees. Each health plan includes various features, such as monthly premium, annual deductible, prescription drug coverage, etc. Due to the number of plans and the number of different features of each plan, the choice between plans becomes difficult for the individual employee. Moreover, the employer typically cannot advise an employee to choose one plan over the other because the employer can be held liable if the employer advises an employee to choose a plan that does not pay for some of the employee's medical expenses. Accordingly, in the employer-sponsored health plan selection process, there exists a long-felt need for methods and systems for facilitating employee choices among health plans.

DISCLOSURE OF THE INVENTION

According to one aspect, the present invention includes a software tool that embodies a "conjoint" model decision process permitting the simplification of difficult choices among complex alternatives into a sequence of short, simpler decisions. "Alternatives" in this context can be products (such as automobiles), services (such as health plans), combinations of complementary services and products, or virtually anything else individuals must decide to choose or not choose. Complex "alternatives" are those defined in terms of many "variables" such that in the decision process a lot of information must be considered. Complex "alternatives" often create difficult decisions that demand that the chooser trade-off the good and bad in each "alternative." For example, the choice between a high-quality bicycle versus a low-quality bicycle, given quality is the only criterion used in the selection, is an easy one. However, as the alternatives become more complex, the choice becomes more difficult and trade-offs must be made. The choice between a high-quality, $500 bicycle that comes in pink only versus a low-quality, $100 bicycle that comes in either green, black, or blue is a more difficult decision than that based on quality only.

The present invention uses, at its core, an adaptation of the conjoint model decision process. The use of the conjoint exercise allows to the tool to assist users in making difficult decisions less complex. By going through the exercise, unique profiles of what is important to the user are developed by the application.

In addition to developing user profiles, the present invention, at the end of the exercise, provides users with a "quality of fit" measure of how well each product or service available to them meets their unique profile.

In order to facilitate user choices among complex alternatives, the present invention includes computer software that requires an individual user to go through a series of less complex choices. The software first presents the user with a list of features. The user selects features which are of importance to the user. The software then presents the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting of each of the selected features. The user is then presented with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes. Each pairing includes a best setting of one attribute and a worst setting of another attribute. The values input by the user in the second series of choices are interpreted as the mathematical difference equal to the relative importance of a best and worst setting of one attribute minus the relative importance of a best and a worst setting for the other attribute in the pairing. A final importance value is calculated for each of the attributes based on the initial relative importance values in the first series of choices and the mathematical difference values. Products and services available to the user are rated based on the final importance values. The user is then presented with data indicating the relative utility to the user of each of the products or services.

Terminology

Before proceeding, a review of keywords and key phrases and their definitions used in this document is warranted. These keywords are placed in double quotes throughout the document to indicate their use may be somewhat different from common use.

Keyword or Key Phrase	Definition
"user"	A person going through the software exercise to gain help in making a choice
"alternative"	A single product or service (among a set of products or services) the "user" can potentially choose
"choice set"	All the "alternatives" the "user" is eligible to choose from
"attribute"	One of numerous variables, each defined as the continuum between its worst "setting" and best "setting," used to define the "alternatives"
"setting" (or "attribute setting")	The value a particular hypothetical or actual "alternative" has for a particular "attribute"; the hypothetical "alternatives" studied during the data-gathering phase of the algorithm are all specified in terms of the worst "setting" vs the best "setting" for each "attribute," whereas actual "alternatives" available to the "user" may be specified by "settings" anywhere along each "attribute's" continuum
"importance"	A measure that the user gives directly via the importance-of-the-difference (between worst and best "settings") screens of the relative importance of a single "attribute"
"difference in importance"	A measure that the user gives directly via the trade-off screens of the (mathematical) difference in the "importance" of two "attributes"
"final computed importance"	A final estimate of the true relative importance of an "attribute" to a "user"
"setting utility" (or "attribute setting utility")	The relative (relative to all "attribute settings") worth or utility of a particular "attribute setting" (anywhere along the "attribute" continuum) to a particular "user"
"total utility"	The total relative (relative to all "alternatives" available to that user) worth or utility of a particular "alternative"; defined as the simple sum across "attributes" of the "setting utilities"

To ensure these keywords and phases are understood, the following example is given.

A person is trying to make a choice between a medium-quality bicycle priced at $250 and a high-quality bicycle priced at $375. The person is given a tool that assists the person in the selection. The tool requires the person to state on a 1-to-5 scale the relative importance of quality and price. For purposes of this example, it is assumed that the person selects 5 and 4, respectively. The values "5" and "4" are "importance" measures as defined above. The tool also asks the person to rate to what degree the person would prefer a high-quality, $500 bicycle to a low-quality, $100 bicycle. In this example, it is assumed that the person indicates a preference for the higher quality, more expensive bicycle, a "+1" on a −4-to-+4 scale. The value +1 is a "difference in importance" value as defined above. The tool then computes that the true importance (on a 1-to-5 scale) of quality and price for this person is a 4.7 and a 4.1, respectively. The values 4.7 and 4.1 are "final computed importance" values, as defined above. These values are used in turn to compute that high-quality is worth 25 (unitless) points to the person whereas medium quality is worth 15 points. The values "15" and "25" are "setting utilities" for the quality and price attributes. Similarly, the tool computes that $250 is worth 15 points to the person and $375 is worth 10. Thus, the tool computes that the total worth of the medium-quality bicycle priced at $250 is 30 points (15+15) and that the total worth of the high-quality bicycle priced at $375 is 35 points (25+10). The 30 and 35 point values are "total utility" values as defined above. Because 35 is higher than 30, the tool has computed that the medium-quality bicycle priced at $250 is worth slightly less to the person, all things considered, than the high-quality bicycle priced at $375. Thus, the tool recommends that the person should choose the high-quality bicycle priced at $375. The following table provides a summary of examples of each keyword or key phrase from the above example.

Keyword or Key Phrase	Example
"user"	The person shopping for a bicycle
"alternative"	The medium-quality bicycle priced at $250 and the high-quality bicycle priced at $375 are the actual "alternatives"; the high-quality, $500 bicycle and the low-quality, $100 bicycle are the hypothetical "alternatives" used in the data-gathering phase
"choice set"	The medium-quality bicycle priced at $250 and the high-quality bicycle priced at $375 together form the actual choice set for the "user"; the high-quality, $500 bicycle and the low-quality, $100 bicycle form a hypothetical "choice set" to which the "user" is asked to react.
"attribute"	Quality is an "attribute" as is price
"setting" (or "attribute setting")	$375 is an actual "setting" of price; $100 is a hypothetical "setting" for price
"importance"	The "5" given for quality
"difference in importance"	The "+1"
"final computed importance"	The "4.7" for quality
"setting utility" (or "attribute setting utility")	The "25" for the high-quality "setting" of quality
"total utility"	The "30" for the medium-quality bicycle priced at $250 "alternative"

One goal in developing the present invention was two-fold. First, create an adaptation of conjoint that is as user friendly as possible (keep it short and easy to understand). Second, go beyond the end of traditional conjoint (developing "utilities") and apply these utilities to the performance of a set of products, presenting the user with a sorted list of how well each product meets their stated preferences. In adapting a research statistical technique, traditionally used to study group preferences, and using it to match individual consumer preferences to actual products or services, the present invention application has established a highly useful product in the marketplace.

As mentioned above, conjoint has been in use since the 1970's. While there are a variety of implementations of the conjoint algorithm in the market the design chosen for implementation of the present invention is exceptionally simple, and yet robust. In addition, the way the present invention has been developed makes it unique. For example, features of real products are evaluated and levels are created, so that all products can be compared by the application in a purely objective basis. The feature attribute descriptions are made as simple and as straightforward as possible. In addition, after the user completes the process of selecting attributes features of the product, making importance of difference decisions, and then trade-off decisions, the application presents results to the user in very simple bar-chart form. The application shows all products available to the user in priority order, based on how well each product matches the preference utility of that individual.

The algorithm utilized by the present invention is unique, in that it provides for paired trade-offs (two by two comparisons of end-point "attribute" characteristics) instead of the usual more complicated trade-offs involving more than

two "attributes" defined not only by their end-points (best and worst "settings") but by numerous "settings" along their entire continuum. As used herein, the term "endpoints" refers to the best and worst settings of an attribute. For example, in the bicycle example discussed above, $100 and $500 are endpoints for the price attribute; whereas "high" and "low" are endpoints for the quality attribute.

According to another aspect of the invention, an "X" matrix utilized to estimate "attribute" "utilities". An "X matrix", as described herein, is a configuration of explanatory variable data, or numbers, in a mathematical format. The "X matrix" designates the independent variable values used in the ordinary least squares matrix set, whereas the "Y Matrix" designates the dependent variable values. Examples of X and Y matrices and their use in calculating utilities for attributes will be discussed in more detail below.

Yet another aspect of the invention is the way in which the results of a user's interaction with the tool are "fed back" to the user. For example, each individual's alternative choices of products, services or concepts are ranked in declining order of "total utility." This way of "reporting" back to each user on how their priorities and decision criteria "value" each alternative clearly indicate the "best fit" choices among all the alternatives in an individual's "choice set."

While one use of the present invention is "attribute" preference-based decision support tool, the software also simultaneously creates databases of user-level "preference data" (i.e. "final computed importance" and "setting utility" data) and other descriptive data. This data has value in the marketplace to producers and middleman organizations as conjoint research and can be used for developing analyses of market share "attribute" importance, and other outcomes of the decision-making process. The creation and merchandizing of this data is very much a fundamental aspect of the tool's value.

Currently, the focus of use for the present invention is in the selection of health plans by consumers or employees of medium-sized or large employers. The fringe benefits of most employers of any size typically include health care financing, and it is not unusual to find medium and large employers offering 3, 4 or more health plans to its employees. An employee of those companies uses the present invention to help him/her select that plan from among the alternatives offered which is best suited to him/her. As discussed above, employees are left by their employers to their own resources in selecting a health plan. The major barrier to a more activist policy by employers is one of liability for employee choices. Another barrier is an economic one since uncertain employees commonly consume huge amounts of human resources staffer time seeking guidance in making their choices among health plans. Left to their own devices, employees will seek advice from other employees or may select a plan based on one or a few criteria—such as premium cost, type of plan or perhaps two or three features of a plan. The present invention alleviates these problems by applying conjoint analysis to facilitate user choices among a large array of complex alternatives.

While the present invention is suitable for assisting employees choose health plans, it has been developed to be generic/flexible enough to be applied to any complex decision. For example, the algorithms described herein are suitable for facilitating user choices among a variety of complex alternative, supplemental insurance, 401k plans/mutual funds, and other product or service categories. Prototypes of this application have been developed, for instance, to assist consumers in choosing computers, and businesses and consumers to choose energy companies. There are many other uses of this application currently being planned.

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiments of the invention will now be explained with reference to the accompanying drawings, of which:

FIG. **1** is a block diagram illustrating an exemplary operating environments for embodiments of the present invention;

FIG. **2** is a block diagram of a selector tool server according to an embodiment of the present invention;

FIG. **3** is a block diagram illustrating the relationship between screens presented to the user by the selector tool server according to an embodiment of the present invention;

FIG. **4** is a screen shot of attribute selection page **302** illustrated in FIG. **3**;

FIG. **5** is a screen shot of importance page **304** illustrated in FIG. **3**;

FIG. **6** is a screen shot of paired trade-off page **306** illustrated in FIG. **3**;

FIG. **7** is a screen shot of results page **308** illustrated in FIG. **3**;

FIG. **8** is a screen shot of details page **310** illustrated in FIG. **3**; and

FIG. **9** is a block diagram of a layered selector tool server according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Overview

A method for facilitating user choices among complex alternatives according to an embodiment of the present invention is comprised of three main steps followed by results and detailed product comparisons. Each of these steps through the exercise are interactive—that is, they engage the "user" and require the "user's" input. In addition, each step relies on the previous step. How the "user" responds in one step will affect what the user is asked to do in subsequent steps. At the end of these steps the user is provided with results.

The main components of the tool are:

1. "Attribute" Selection—branded on the site displayed as "Attribute Selection"
2. "Importance" Ratings—branded on the site displayed as "Importance of Difference"
3. "Difference in Importance" Ratings—branded on the site displayed as "Trade-Offs"
4. Results
5. Detailed "Alternative" Comparisons

Software for implementing each of these steps will be discussed in more detail below following a discussion of the operating environment for the software.

Exemplary Operating Environment

Turning to the drawings, wherein like reference numerals refer to like elements, the invention is illustrated as being implemented in a suitable computing environment. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by a personal computer.

Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

With reference to FIG. **1**, an exemplary system for implementing the invention includes a general purpose computing device in the form of a conventional personal computer **20**, including a processing unit **21**, a system memory **22**, and a system bus **23** that couples various system components including the system memory to the processing unit **21**. The system bus **23** may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) **24** and random access memory (RAM) **25**. A basic input/output system (BIOS) **26**, containing the basic routines that help to transfer information between elements within the personal computer **20**, such as during start-up, is stored in ROM **24**. The personal computer **20** further includes a hard disk drive **27** for reading from and writing to a hard disk, not shown, a magnetic disk drive **28** for reading from or writing to a removable magnetic disk **29**, and an optical disk drive **30** for reading from or writing to a removable optical disk **31** such as a CD ROM or other optical media.

The hard disk drive **27**, magnetic disk drive **28**, and optical disk drive **30** are connected to the system bus **23** by a hard disk drive interface **32**, a magnetic disk drive interface **33**, and an optical disk drive interface **34**, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for the personal computer **20**. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk **29**, and a removable optical disk **31**, it will be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories, read only memories, and the like may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk **29**, optical disk **31**, ROM **24** or RAM **25**, including an operating system **35**, one or more applications programs **36**, other program modules **37**, and program data **38**.

A user may enter commands and information into the personal computer **20** through input devices such as a keyboard **40** and a pointing device **42**. Other input devices (not shown) may include a microphone, touch panel, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit **21** through a serial port interface **46** that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port or a universal serial bus (USB). A monitor **47** or other type of display device is also connected to the system bus **23** via an interface, such as a video adapter **48**. In addition to the monitor, personal computers typically include other peripheral output devices, not shown, such as speakers and printers. With regard to the present invention, the user may use one of the input devices to input data indicating the user's preference between alternatives presented to the user via monitor **47**.

The personal computer **20** may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer **49**. The remote computer **49** may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the personal computer **20**, although only a memory storage device **50** has been illustrated in FIG. **1**. The logical connections depicted in FIG. **1** include a local area network (LAN) **51**, a wide area network (WAN) **52**, and a system area network (SAN) **53**. Local- and wide-area networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

System area networking environments are used to interconnect nodes within a distributed computing system, such as a cluster. For example, in the illustrated embodiment, the personal computer **20** may comprise a first node in a cluster and the remote computer **49** may comprise a second node in the cluster. In such an environment, it is preferable that the personal computer **20** and the remote computer **49** be under a common administrative domain. Thus, although the computer **49** is labeled "remote", the computer **49** may be in close physical proximity to the personal computer **20**.

When used in a LAN or SAN networking environment, the personal computer **20** is connected to the local network **51** or system network **53** through the network interface adapters **54** and **54A**. The network interface adapters **54** and **54A** may include processing units **55** and **55A** and one or more memory units **56** and **56A**.

When used in a WAN networking environment, the personal computer **20** typically includes a modem **58** or other means for establishing communications over the WAN **52**. The modem **58**, which may be internal or external, is connected to the system bus **23** via the serial port interface **46**. In a networked environment, program modules depicted relative to the personal computer **20**, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

In the description that follows, the invention will be described with reference to acts and symbolic representations of operations that are performed by one or more computers, unless indicated otherwise. As such, it will be understood that such acts and operations, which are at times referred to as being computer-executed, include the manipulation by the processing unit of the computer and/or the processing units of I/O devices of electrical signals representing data in a structured form. This manipulation transforms the data or maintains it at locations in the memory system of the computer and/or the memory systems of I/O devices, which reconfigures or otherwise alters the operation of the computer and/or the I/O devices in a manner well understood by those skilled in the art. The data structures where data is maintained are physical locations of the memory that have particular properties defined by the format of the data. However, while the invention is being described in the foregoing context, it is not meant to be limiting as those of skill in the art will appreciate that the acts and operations described hereinafter may also be implemented in hardware.

In FIG. 1, exemplary application programs **36** used to implement the present invention include a selector tool server **36A** implemented on local computer **20** and a selector tool client **36B** implemented on remote computer **49**. Selector tool server **36A** may communicate with selector tool client **36B** over local area network **51**, system area network **53**, or wide area network **52**. Exemplary communication protocols that may be used for communication between selector tool server **36A** and selector tool client **36B** over LAN **51** or SAN **53** include HTTP over TCP/IP. Exemplary communication protocols that may be used to communicate between selector tool server **36A** and selector tool client **36B** over WAN **52** include the point-to-point protocol (PPP).

Selector tool server **36A** preferably performs the functions of presenting the user with a series of choices relating to complex alternatives, calculating relative utility scores of the alternatives based on the choices, and feeding back to the user an indication of which of the complex alternatives has the highest utility for that user. Selector tool client **36B** may establish a connection with selector tool server **36A** in order to provide communications between the user and selector tool server **36A**. In a preferred embodiment, selector tool client **36B** comprises a web browser, such as Internet Explorer available from Microsoft Corporation of Redmond, Wash., or Netscape Navigator available from America Online Corporation of Reston, Va.

The present invention is not limited to a selector tool that is implemented as a selector tool server and a selector tool client connected via a network. For example, in an alternative embodiment, the present invention may be implemented entirely on a local machine wherein the selector tool comprises an application program that presents the user with a series of choices relating to complex alternatives, calculates the relative utility of the complex alternatives, and feeds the relative utility information back to the user entirely on the local machine. However, a networked environment is preferred so that multiple users can access the selector tool server. For example, for company health plan selection, selector tool server **36A** may be resident on a server accessible by company employees. In this manner, all employees may access selector tool server **36A** using a web browser that is common on most personal computers. In addition, when the health plans offered by a company change, it is easier to update a server or a set of servers than it is to update software on each individual user's personal computer.

FIG. **2** is a block diagram of selector tool server **36A**. In the illustrated embodiment, selector tool server **36A** includes a user interface generator **200** and a utilities calculation engine **202**. User interface generator **200** may be a web server. User interface generator **200** preferably presents a series of input screens to the user relating to choices between complex alternatives. User interface generator **200** also receives input from the user and delivers that input to utilities calculation engine **202**. Utilities calculation engine **202** calculates a total relative utility value for each of the complex alternatives, based on the user choices. Utilities calculation engine **202** outputs the total utilities calculations to user interface generator **200**. User interface generator **200** then outputs the total utilities values to selector tool client **36B**.

FIG. **3** is a block diagram of exemplary screens that may be presented by user interface generator **200** illustrated in FIG. **2**. In FIG. **3**, selector tool instruction page **300** provides an overview of how the selector tools works, steps required to be performed by the user in using the tool, and the length of time required to complete the selection. Because selector tool instruction page **300** does not provide an functionality important to describing the invention, further description relating thereto will not be presented herein.

Attribute selection page **302** provides a listing of attributes, allows the user to select attributes that are of importance to the user, and links each attribute to terms to know. FIG. **4** is a screen shot of an exemplary attribute selection page **302**. In FIG. **4**, attribute selection page **302** includes an instruction portion **400** that provides the user with instructions on how to use attribute selection page **302**. In the illustrated example, instruction portion **400** instructs the user that the user must select at least four attributes in order for the tool to work. Attribute selection page **302** also includes an attribute selection portion **402** that allows the user to select attributes that are of importance to the user and that allows the user to access definitions of terms. In the illustrated example, attribute selection portion **402** includes a table containing attribute categories relating to employer-sponsored health plans, such as cost and access. Each attribute category includes a series of attributes. In the illustrated example, the "cost" attribute category includes attributes such as monthly employee contribution, annual deductible, separate per-admission hospital deductible, etc. Similarly, the "access" attribute category includes attributes such as coverage of brand name prescription drugs of choice and the ability to self-refer to a specialist. In order to select an attribute of importance, the user clicks on the appropriate box using an input device, such as a mouse. The attributes selected by the user will be used in calculating the relative utilities of health plans.

Referring back to FIG. **3**, once the user has selected attributes that are of importance to the user, the user is presented with importance page **304**, which allows the user to rate the importance of each attribute. FIG. **5** is a screen shot illustrating an exemplary embodiment of importance page **304**. In FIG. **5**, importance page **304** includes an instructions portion **500** that explains to the user the mechanics of using importance page **304**. Importance page **304** also includes an importance of difference portion **502** that presents the user with two hypothetical values that a plan could possess for each attribute: a high value and a low value. A scale is provided that requires the user to rate how important the difference is to the user between the two possible alternatives. In the illustrated example, the attribute being evaluated is monthly employee contribution to a health plan. The user is asked to rate the importance of the difference between a plan with an employee contribution of $38 per month versus a plan with an employee contribution of $90 per month. A scale is provided below the choice that allows the user to rate the degree of importance between the two settings for each variable selected: a best setting and a worst setting. In the illustrated example, the user may indicate that the difference between an employee contribution of $38 per month and an employee contribution of $90 per month is very important. Importance page **304** stores a value indicative of the relative importance of the difference to the user and passes this value to utilities calculation engine **202** (illustrated in FIG. **2**).

The user is preferably presented with a series of preference pages **304** that require the user to rate the relative user's preference between best and worst settings of variables relating to an attribute. Once the user has completed the importance of difference rating step, referring back to FIG. **3**, the user is presented with a series of paired trade-off pages **306**. Each paired trade-off page **306** requires the user to rate paired sets of attributes. In particular, the user is presented with the best setting of one attribute and the worst setting of another attribute versus the worst setting of the first attribute

and the best setting of the second attribute and the user is asked to rate the relative degree of importance of the best and worst settings of the different attributes.

FIG. 6 illustrates an exemplary paired trade-off page **306** according to an embodiment of the present invention. In FIG. **6**, paired trade-off page **306** includes instructions portion **600** that instructs the user on how to select between the alternatives presented on page **306**. Paired trade-off page **306** also includes a trade-off selection portion **602** which presents the user with two pairings of two different attributes. In a preferred embodiment, the user is presented with a first pairing that includes a highest setting of one attribute and a lowest setting of another attribute and a second pairing that contains the highest setting of one attribute and the lowest setting of another attribute. For example, if the attributes are A1 and A2, the first pairing would be high(A1) AND low(A2), and the second pairing would be low(A1) AND high(A2). In the example illustrated in FIG. **6**, the pairings of attributes are as follows: an annual deductible of $0 and an employee contribution of $90 per month versus an annual deductible of $500 and an employee contribution of $38 per month. The represent combinations of high and low settings of the illustrated attributes—monthly contribution and annual deductible. The user is required to rate the user's preference between the pairings using discreet values on the scale provided below the pairings. For example, if the user strongly prefers an annual deductible of $500 and an employee contribution of $38 per month versus an annual deductible of $0 and an employee contribution of $90 per month, the user may select one of the circles on the right side of pair trade-off page **306**.

The user is presented with a series of paired trade-off screens **306** and is required to rate the user's preference for each of the pairings. Values indicative of each of the user-selected ratings are provided to utilities calculation engine **202** illustrated in FIG. **2**.

Once the user has completed all of the importance pages **304** and paired trade-off pages **306**, the selector tool performs the following steps:

1. Using utilities calculation engine **202**, a "final computed importance" is calculated for each "attribute" used in the exercise. This is a measure of the relative importance of the "attribute" with respect to the "settings" of that "attribute" presented as well as to the relative importance of all the other "attributes."
2. Utilities calculation engine **202** gathers the performance levels (actual "settings") for each "alternative" available to the "user," for each "attribute."
3. For each "alternative" and "attribute" utilities calculation engine **202** creates a "setting utility" score for each "alternative" available to the "user." Details of this calculation are described below. The sum of the "setting utility" scores for all the "attributes" for an "alternative" become the "alternative's" "total utility" score. This value is unique to each "alternative"/"user" combination.
4. The "user" is then presented with a list of all the "alternatives" available to them with a graphical representation of the relative "total utility" scores of the "alternatives." With this information, the user can ascertain how well each "alternative" matches their stated importance.

Referring back to FIG. 2, results page **308** presents the results from the utilities calculation to the user. The utilities are measured in terms of a preference value that indicates the relative utility of each choice to that particular user. Alternative choices may be sorted in any order, such as descending order. FIG. 7 illustrates an exemplary results page **308** according to an embodiment of the present invention. In FIG. **7**, results page **308** includes a description portion **700** that describes the contents of results page **308**. Importance indicator portion **702** includes a table having a bar graph that graphs the importance indicator value for each of the complex alternatives. In the illustrated example, the bar graph indicates that PSC test plan 2 more closely matches the user's preferences than PSC test plan **1**. Accordingly, based on results page **308**, the user should select PSC test plan **2** as the user's health plan.

Referring back to FIG. **3**, a details page **310** can be accessed from any of the other pages **300, 302, 304, 306**, and **308**. Details page **310** explains the feature of the page from which details page **310** was accessed. For example, if details page **310** is accessed from results page **308**, a more comprehensive explanation of the results will be presented. FIG. **8** illustrates an example of details page **310** accessed from results page **308**. Referring to FIG. **8**, details page **310** includes an explanation portion **800** that explains the content of details page **310**. Details page **310** includes a content table **802** that compares the attributes of two choices presented to the user. In the illustrated example, the choices are employer-sponsored health plans. The attributes include features of the health plans, such as annual deductible, employee contribution, and out-of-pocket costs for prescriptions.

Thus, as illustrated above, the present invention provides a user-friendly graphical user interface that simplifies complex choices for users. The users are presented with two series of simple choices. Based on the user's responses to the simple choices, the utilities calculation engine of the present invention calculates the relative utility of complex choices for the user. By breaking the complex choices into series of simple choices, the present invention greatly facilitates user selection among complex choices.

Detailed Description of Calculations Performed by Utilities Calculation Engine

The statistical algorithm implemented by utilities calculation engine **202** involves the calculation of "regression" coefficients (u_k) for the equation:

$$y_i=\alpha_i+u_1*a_{1i}+u_2*a_{2i}+\ldots+u_k*a_{ki}+e_i$$

where y_i is a variable with i possible observations per tool user, representing the quantitative values chosen by a individual to measure his or her judgment of the importance of the difference between some best value of each important "attribute" and some worst value of the "attribute", as well as values input by that individual to measure his degree of attraction to either of two two-"attribute" alternatives (or choices) (in which the key characteristic of those two-"attribute" alternatives is that the first "attribute" has its best possible value while the second has its worst possible value, and the other alternative's two "attributes" have the characteristic that the first "attribute" has its worst possible value while the second "attribute" has its best possible value). It is important to note that while the "user" provides the relative degree to which he/she prefers one "best/worst" "alternative" (i.e. product) to the other, the algorithm for deriving "final estimated importances" (i.e. the regression analysis) interprets these responses as the mathematical difference in the "importances" of the two "attributes." This unique property of the algorithm is so important that the following example is warranted to clearly illustrate the mechanics.

Assume the two "attributes" of a trade-off screen **306** are price and quality and assume the "alternatives" are bicyclesThus, the trade-off screen might look like the following: .

$500 High Quality	vs.	$100 Low Quality
−4 −3 −2 −1	0	+1 +2 +3 +4

The user is asked to respond with a number between −4 and +4, with a −4 meaning he/she strongly prefers the High Quality, $500 bicycle and a +4 meaning he/she strongly prefers the Low Quality, $100 bicycle. Values between +4 and −4 indicate less strong preferences, with 0 (zero) indicating the user prefers the two bikes equally. Because all "attributes" are studied in the trade-off screens only in terms of their best and worst "settings" and also because exactly two "attributes" are studied in each trade-off screen **306**, any value on the +4 to −4 continuum has a second interpretation, besides the relative preference of one "alternative" vs. the other, which is the mathematical difference in the "importance" of the two "attributes" being studied (where the "importance" values are on a +1 to +5 scale). Thus, if a user states a "+3" in the above trade-off, that can be interpreted to mean that price is more "important" than quality and precisely that the "importance" of price is 3 more "importance" points than the "importance" of quality. In other words, the user may have "importance" values for price and quality of +4 and +1 respectively, or +5 and +2 or +4.5 and +1.5 (the actual importance magnitudes are gauged elsewhere (not in the trade-off screens), in the "importance" (i.e. importance of difference) screens). Other trade-off values have similar interpretations (e.g. a "0" is interpreted as the "importance" of price and quality being equal; a "−4" is interpreted as the "importance" of price being 4 "importance" points less than the "importance" of quality (e.g. price "importance"=1 and quality "importance"=5). Another way to view this is that while the user is responding (with respect to moving from −4 to +4) in terms of left-of-screen "alternative" vs. right-of-screen "alternative," the software algorithm interpretation is bottom-of-screen "attribute" vs. top-of-screen "attribute"." This property (interpretation) of the algorithm may indeed be the single most unique aspect of utilities calculation engine **202**.

The present invention is not limited to using any particular scale for rating user preferences with regard to difference in importance or trade-offs. The size and increments in the scale depend on the desired granularity and the algorithm used to generate the total utility value.

As described above, two types of data are gathered from users:

1. The importance of a single "attribute" (defined as the importance of the difference between the best and worst "settings" of that "attribute"), measured on a 1-to-5 scale (using importance pages **304**).
2. The difference in the importance of two "attributes. Since the "importance" of each "attribute" is measured on a 1-to-5 scale (see 1 above), the "difference in importance" is measured on a −4-to-4 scale (since the minimum value is a 1 minus a 5 which equals −4 and the maximum value is a 5 minus a 1 which equals 4) (using paired trade-off pages **308**).

Via regression analysis, both types of data are analyzed together as a single set of information. The result of this regression analysis is a single number for each "attribute" indicating the final estimate of how important the user feels the "attribute" is (on the original "importance" scale).

The way the data is coded is as follows:

Dependent Variable Vector Y

Type 1 "importance" data is simply bottom-augmented with the type 2 "difference in importance" data. Thus, if a user had been asked about 5 "attributes" and had given "importance" scores of 5, 4, 3, 2, and 1, the Y "importance" data would equal:

5
4
3
2
1

Similarly, if the user had seen 6 pairs of "attributes" and had given "difference in importance" ratings of −4, 3, 0, 1, 0, and −2, the Y "difference in importance" data would equal:

−4
3
0
1
0
−2

Thus, the final Y vector would equal:

5
4
3
2
1
−4
3
0
1
0
−2

The number of rows equals the number of observations of data used in the regression analysis and are equal to the number of separate data values supplied by the user for "importance" and for "difference in importance" questions.

Independent Variable Matrix X

The X matrix associated with Type 1 "importance" data is made of rows that are "dummy" coded with a "1" indicating the "attribute" the user is referring to and 0's otherwise. Thus, assuming the "importance" scores of 5, 4, 3, 2, and 1 given above were in the order of first, second, third, fourth, and fifth "attributes, the X "importance" data would equal:

1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

Note that it is not necessary that the order is first, second, third, fourth, and fifth, however. If the data had been in, say, order of second, first, third, fourth, and fifth attributes, then the X matrix would have looked like:

0 1 0 0 0
1 0 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

For this example, the former order is assumed.

In addition to determining an X-matrix for the importance data entered through importance pages **304**, utilities calculation preferably also determines an X-matrix for difference of importance data collected through paired trade-off screens **306**.

The X matrix associated with the "difference of importance" responses is made of rows that are coded as "1" if the "attribute" is one of the pair being traded off and is the "attribute" shown either at the top or the bottom of the screen, "-1" if the "attribute" is one of the pair being traded off and is the "attribute" shown opposite the other "attribute" (at bottom if former is at top of screen and vice versa), and "0" if the "attribute" is not one of the pair being traded off. For example, assuming that the "top-bottom" pairs of "attributes" are first-second, second-third, third-fourth, fourth-fifth, fifth-first, and fourth-second, and the top "attribute" has its best "setting" on the right side of the screen and bottom "attribute" has its best "setting" on the left side of the screen (though this may be interchanged), the X matrix associated with the "difference of importance" data equals

1	-1	0	0	0
0	1	-1	0	0
0	0	1	-1	0
0	0	0	1	-1
-1	0	0	0	1
0	-1	0	1	0

The above X matrix is given only as an example. Pairings of "attributes" may be random or a systematic approach may be used to pair the attributes, as in an orthogonal or near-orthogonal design.

As before with the Y vector, the final X matrix is created by bottom-augmenting the "importance" data with the "difference in importance" data. Thus the final X matrix equals

1 0 0 0 0
0 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
1 -1 0 0 0
0 1 -1 0 0
0 0 1 -1 0
0 0 0 1 -1
-1 0 0 0 1
0 -1 0 1 0

Thus, the final set of data to be analyzed is as follows:

Y vector	X matrix
5	1 0 0 0 0
4	0 1 0 0 0
3	0 0 1 0 0
2	0 0 0 1 0
1	0 0 0 0 1
-4	1 -1 0 0 0
3	0 1 -1 0 0
0	0 0 1 -1 0
1	0 0 0 1 -1
0	-1 0 0 0 1
-2	0 -1 0 1 0
2	0 0 0 1 0
1	0 0 0 0 1
-4	1 -1 0 0 0
3	0 1 -1 0 0
0	0 0 1 -1 0
1	0 0 0 1 -1
0	-1 0 0 0 1
-2	0 -1 0 1 0

In summary, each row (of both the Y vector and the X matrix) represents a single response the "user" has given, the actual response being recorded in the Y vector. Each column of the X matrix represents an "attribute" the "user" has chosen as important (where rows with only 1's and 0's have a 1 in the column of the "attribute" that is the subject of the "importance"-of-difference screen and rows with 1's, 0's, and −1's have a 1 or −1 in a column to identify which attribute was at the top of and which attribute was at the bottom of the trade-off screen.

From this type data set, the B regression coefficients are calculated via

$$B=(X'X)^{-1}X'Y,$$

where $^{-1}$ indicates the matrix inversion transformation and the single quote (') indicates the matrix transpose transformation, are directly interpretable as importance (on the 1-to-5 "importance" scale) scores for the "attributes" they modify. Each B coefficient is paired with and thus modifies one "attribute." A slight variation of this form is to include an additional column of ones ("1"s) in the leftmost position of the X matrix. The extra B coefficient in that instance is not associated with an "attribute" but is a measure of the intercept of the regression model, which theoretically equals zero. That is, in the absence of predicting anything about importance (e.g. an "importance" by plugging in a "1" or a "difference in importance" by plugging in a "1" and a "−1"), that is, only a series of zeros are plugged in, the model predicts zero. That is to say that the null condition of using the model to do nothing "predicts" a "0" (zero) "importance."

Each user chooses which m or more "attributes" he/she actually uses in making a selection, where m is a variable threshold and is set in advance. If this number exceeds n then the top n (based on the user's "importance" scores (and predetermined expert judgments of the importance of "attributes" if a tie-breaker is needed)) are used and it is only these that have "final computed importances" calculated via the above formula. Again, n is a variable threshold that is set in advance. For excluded "attributes, the "final computed importance" is set to zero. Thus, the number of "attributes" for which "final computed importances" are calculated varies from user to user. Currently, the software is written to allow for "importance" scores to be calculated for anywhere from four to fifteen "attributes" (that is, m=4 and n=15).

While it would be sufficient to simply use the (direct) "importance" scores for each "attribute" to help users select products/services, it is believed that the augmented data, the "difference in importance" data, greatly improves "final computed importances" and thus recommendation results. This is so because users are not actually told they are providing "difference in importance or trade-off" scores, but rather are told to gauge their preference for the best "setting" of one "attribute" paired with the worst of the other versus the worst "setting" of the one "attribute" paired with the best of the other. In this way, the user is actually comparing two products or services. Because only the best and worst "settings" are ever shown and because only two "attributes" are shown at a time, the interpretation of the responses is precisely "the mathematical difference in the importance of the two 'attributes' (each on a 1-to-5 scale)." This indirect way of obtaining information on the relative impact importance of "attributes" by comparing a series of hypothetical "alternatives" is a more realistic task for the "user" and is believed to enhance relative importance measurement.

With "final computed importances" estimated/calculated, the algorithm moves to using that data to calculate "total utility" scores for each product or service the user is in the market to buy and is eligible to buy. For each "attribute", the "final computed importance" (i.e. the B coefficient from the "attribute") is taken as the value of the worst "setting" of the "attribute" and five times the "final computed importance" is taken to be the value of the best "setting" of the "attribute". Then, using linear interpolation, the "setting" or specification of each product or service is transformed to a value to be presented to the user. For example, if the "final computed importance" of price is 3.2 and the worst price "setting" is \$50, and the best price "setting" is \$10, then a product with a price of \$23.33 would have a "setting utility" for price of

$$3.2+(5*3.2-3.2)*((23.33-50)/(10-50))=11.7344.$$

The utility numbers are "unitless"; 10.6656 is not 10.6656 dollars or anything else. They are relative numbers that allow for comparing specifications on different "attributes" for a given user. A "setting" on an "attribute" that has a "10" value for user is worth more than a "setting" on the same or another "attribute" that has a value of, say, "7." Individual "attribute" values are summed across the "attributes" making up the products or services, yielding a "total utility" score for each product or service for each user.

Utilities calculation engine **202** then rank orders the products or services for the user, in descending order of "total utility." The top product or service in this list is the best recommendation for the product or service the user should choose, given the relative importance the user places on the various "attributes" that make up the products or services and the actual specifications of the products or services.

Although the present invention can be used to calculate relative total utilities of products or services based on any number of attributes and attribute pairings, the following examples illustrate exemplary numbers of attributes and corresponding attribute pairings suitable for use with the present invention.

(1) 4 "attribute" engine

Pairs	Interpretation
1,4 2,3 3,4 4,2 3,1 1,2	This array of paired integers describes the comparisons of combinations of "attributes" made by utilities calculation engine 202. The pairings listed here are for 4 attributes and are believed to be unique. The remaining pairings listed herein are for 5 to 15 attributes and are pairings listed herein are for 5 to 15 attributes and are also believed to be unique.

(2) 5 "attribute" engine

1,2
2,4
5,3
3,1
1,4
3,4
1,5
5,2

(3) 6 "attribute" engine

1,2
2,4
5,3
3,1
6,4
1,6
3,4
2,5

(4) 7 "attribute" engine

1,2
2,4
5,2
6,7
3,1
7,3
1,6
3,4
4,5

(5) 8 "attribute" engine

1,2
5,8
2,4
5,2
6,7
3,1
7,3
1,6
3,4
8,6
4,5

(6) 9 "attribute" engine

1,2
9,7
2,4
7,3
4,5
6,1
5,2
8,6
3,1
8,9
4,3

(7) 10 "attribute" engine

1,2
9,7
2,4
7,3
4,5
10,9
6,1
5,2
4,3

-continued

8,6
3,1
8,10
(8) 11 "attribute" engine
1,2
9,7
2,4
7,3
4,5
10,11
6,1
5,2
4,3
11,9
8,6
3,1
8,10
(9) 12 "attribute" engine
1,2
9,7
2,4
10,12
7,3
4,5
12,11
6,1
5,2
11,9
4,3
8,6
3,1
8,10
(10) 13 "attribute" engine
11,13
1,2,
9,7
2,4
10,12
7,3
4,5
12,11
6,1
5,2
11,9
4,3
8,6
3,1
8,10
13,9
(11) 14 "attribute" engine
1,13
1,2
9,7
2,4
10,12
7,3
4,5
12,11
6,1
5,2
14,12
11,9
4,3
10,14
8,6
3,1
8,10
13,9
(11) 15 "attribute" engine
11,13
1,2
8,15
9,7
2,3

-continued

10,12
7,3
4,5
12,11
6,1
14,12
11,0
4,3
10,1
8,6
15,14
3,1
8,10
13,9

In all the above, the numbers in the pairs (e.g. "13,9") signify the two attributes in terms of the rank order of their "importance" ratings by the user. For instance, "13,9" signifies the 13^{th} most important attribute and the 9^{th} most important attribute.

Criteria Matching Heuristic

According to another aspect, the present invention includes a criteria matching heuristic for determining alternatives available to a user based on information received from the user.

One of the critical issues in applying a decision tool to the selection of an "alternative" is ensuring the "user" is presented with those "alternatives" that are available to him/her in the marketplace. Especially within the realm of health care plan choice, plans are often times available only based on specific, and sometimes complicated criteria (zip code, employment status, etc). In addition, not only does the "user" need to be "matched" to those product "alternatives" available to them, but they must also be presented with the "attributes" and "attribute settings" that apply to them. For example, users" looking for health plan coverage for a family will need to see a different set of premiums than do users looking for self-coverage. These two pre-requisites, match users to the appropriate "alternatives" and "attribute settings," are critical to the goal of providing the "user" with an accurate and useful end product.

The present invention, in attempting to solve these requirements, includes a unique criteria matching solution. This solution relies on a data driven mechanism that automates and "genericizes" the process of matching "users" to "alternatives" and "attribute/attribute settings," regardless of the "alternative" category and "attributes" involved. The criteria matching system works as follows:

1) Specific criteria, which may be used to link a "user" to an "alternative" or "attribute," are created. This could include such things as "single" or "family" or "full time," etc. These criteria are populated in a database table.
2) The "attributes" to be used in the project are also populated in a separate database table.
3) A "join" table is created that links which criteria apply to each of the "alternatives."
4) An "alternative" can have multiple criteria associated with it, while a criteria can be associated with multiple "alternatives."
5) The "alternatives" to be used in the project are populated in a separate database table.
6) A "join" table is created that links the criteria that apply to each of the "attributes."

7) An "alternative" can have multiple criteria associated with it, while criteria can be associated with multiple "attributes."

Part of the selector tool described above is the "registration." This is the initial step that requires the user to input criteria determining questions, such as "What is your zip code," etc. These registration questions are also data driven—that is, the questions and the responses are drawn from one data table and are written to another data table. The answers to these registration questions are linked, via another join table, to the criteria that have been established for the project.

Once the user answers registration questions, the responses trigger a look up of which criteria apply to the user. Using this criteria and the criteria-"alternative" and criteria-"attribute" join tables, the user is automatically matched with the "alternatives" and "attributes" that are applicable to them. One example would be the question, "are you a member of any of the following unions? APWU, Mailhandlers or Postal Workers. If the user answers "yes," and selects APWU, then the application would only match the user's preferences to plans which are available to members of this union and with the benefit structure of that plan which is only available to members of that union. Another example might be a user's answer to the questions, 1) what is your age, and 2) are you eligible for Medicare? Again, if the user answers "Yes," the user will only be shown plans to which that user is eligible and at the rates that are applicable to person's eligible for Medicare.

This structure has been developed so that it is "alternative"-category and "attribute" in-specific; it is a highly flexible methodology for meeting the purpose of joining users to "alternatives" and "attributes."

Criteria Matching Heuristic Example

The following is an abbreviated example of how the criteria matching heuristic works for a health plan selection tool according to an embodiment of the present invention. In this example, the following assumptions are made:

There are 2 types of employees, Union 1 and Union 2.

For any given plan, premium varies by coverage level as well as the union to which the employee belongs.

Employees can choose from 2 types of coverage levels—individual coverage and family coverage.

There are 6 possible health plans:

2 available in NY zip codes (Plan A, Plan B)

2 available in NJ (Plan C, Plan D)

1 available nationally (Plan E)

1 available regardless of coverage area, but the employee must belong to Union 1 (Plan F)

The following questions can be presented to each employee through a graphical user interface:

Question 1: What Union do you belong to?

1. Union 1
2. Union 2

Question 2: What type of coverage level are you looking for?

1. Individual
2. Family

Question 3: What is your zip code?

The following tables are used by the selector tool to match users with attributes and products. Each of these tables may be stored in a computer-readable medium, such as a memory device.

TABLE 1

Criteria - Defines the criteria needed for the project to match users with attributes and products (such as health plans). Note the ZipCodeExclusive flag may be used to prevent certain user groups from being matched to products based on coverage area (zip code) if need. The ZipCodeExclusive flag is not used in this example.

Criteria_ID	Description	CriteriaType	ZipCodeExclusive
1	All Users	Ubiquitous	N
2	Union 1	Ubiquitous	N
3	Union 2	Ubiquitous	N
4	Union 1, Individual	Attribute	N
5	Union 1, Family	Attribute	N
6	Union 2, Individual	Attribute	N
7	Union 2, Family	Attribute	N
8	Individual	Attribute	N
9	Family	Attribute	N

TABLE 2

ProductCriteria - Defines which plans are matched via criteria. There are two steps to matching users to products (plans). Step one is by association with criteria. Step two is by zip code coverage.

Criteria_ID	Product_ID
2	6

TABLE 3

ProductAttribute - Defines the attributes for a project

ProductAttribute_ID	Description
1	Premium
2	Annual Deductible
3	Chiropractic

TABLE 4

Product - Defines the products (such as health plans) for a project

ProductAttribute_ID	Description
1	Plan A
2	Plan B
3	Plan C
4	Plan D
5	Plan E
6	Plan F

TABLE 5

AttributeOption - Defines the possible variants of attributes for a project. For example, while there is one attribute "annual deductible," there are 2 sets of values for each plan - the individual value and the family value. Depending on the desired coverage level, users should be matched to the appropriate plan values.

ProductAttribute_ID	Criteria_ID	Description
1	4	Premium, Union 1, Individual
1	6	Premium, Union 2, Individual
1	5	Premium, Union 1, Family
1	7	Premium, Union 2, Family
2	8	Annual Deductible, Individual
2	9	Annual Deductible, Family
3	1	Chiropractic

TABLE 6

ProductCoverage - Associates Products with zip codes. In Table 6, the zip codes have been abbreviated as "new york zips" and "new jersey zips" for purposes of this example. In a real project, for each product, a record would exist for each zip code associated with the product.

Zipcode	Product_ID
(new jersey zips)	1
(new jersey zips)	2
(new york zips)	3
(new york zips)	4
(both new york and new jersey zips)	5

Steps for Performing Criteria Matching Heuristic

1. The user is asked to answer the registration questions. Depending on how the user answers the questions, criteria are associated with each user type.
2. Once users are associated with criteria, joins are done to determine which products are available to the user. This is a 2-step process. In step 1 users are matched with plans that are only associated by criteria (not coverage area (zip code)). The ProductCriteria table (Table 2) is used for this purpose.
   In step 2 users are matched to products by coverage area (zip code). The ProductCoverage table (Table 6) is used for this purpose.
3. In addition, to joining the user with products, users are also matched to their appropriate attributes; that is, the attributes that are available and pertinent to their decision, based on the registration questions.
4. Once these joins have been done, the user can proceed with using the tool.

Sample Answers and Results

1. The user answers Q1 "Union 1," Q2 "individual" and Q3 as a New Jersey zip code.
2. 4 criteria are associated with the user:
   1, 2,4 and 8
3. Step 1 of matching the user with the products available to the user is to see if any of the user's criteria meet the ProductCriteria conditions. In this case, the user would be eligible for product_id **6** (Plan F) since the user's criteria of 2 matches a record in the ProductCriteria table (Table 2).
4. Step 2 of matching the user with the products available to the user is to see which plans, if any, are available to the user due to the user's zip code. In this case product_ids **1** (Plan A), **2** (Plan B) and **4** (Plan E) would be available to the user based on the zip code that the user entered.
5. Now the user is matched with attributes. Given the user's criteria, the user would be matched with the ProductAttribute_id/Criteria_id combinations of **1**,**4** (Premium, Union **1**, Individual) **2**, **8** (Annual deductible, Individual) and **3**, **1** (Chiropractic, all users). With this identified, the tool can ensure the user trades off between the relevant attributes, and that the attribute values used for each plan are appropriate for the user.

Other Applications

Although the invention has been described above with regard to health plan selection, the present invention is not limited to such an embodiment. The selector tool software according to the present invention can be modified to assist users in almost any complex decision. Examples of additional applications of the selector tool software are discussed in detail below.

At the core of the selector tool software are seven steps:

1. Create a list of product or service attributes, which can be objectively collected for all products or services offered. Create levels for these attributes so their values can be compared.
2. Allow users of the software to simply click on those attributes or features, which are important to them.
3. For each attribute selected, show the high and low levels for each, and ask the user to determine how important the difference between the extremes are in their selection process.
4. Using the algorithms discussed above, present the user with unique combinations (paired comparisons) of attributes which they have indicated as very important, and force them to make trade-off decisions between hypothetical plans/products which contain these features.
5. Present the results of their preference exercise by showing them all products available to them in priority order, based on how well each product fits the user's profile
6. Allow the user to access a "Compare" feature, which allows the user to compare any four plans or products; side-by-side, feature-by-feature
7. Allow the user to purchase or enroll in whatever they choose.

One additional application of the invention is to assist patients and physicians in making complex medical decisions. For instance, the selector tool can be used to help patients choose between angioplasty or bypass. It can be used to help patients choose between radical mastectomy or lumpectomy or choosing between drugs.

Another application for the present invention is a virtual on-line brokerage firm, initially offering health plans to small businesses nationwide.

Another application of the invention is for a company which can broker energy company services in de-regulated states. There are already more than 15 states which are deregulated and more are moving in this direction. The selector tool will allow businesses and consumers to click on what is important to them, make trade-offs (for instance between cost, comfort, green-power, other bundled services, etc.) and have the software tell them which plan fits their needs best.

Yet another application for the selector tool is for an Internet-based recruiting business. Such a business would be tremendously empowered through use of the selector tool technologies. With monster.com, for instance, searching is done based on simple questionnaires. The results are simply a list, in no priority order, of resumes or positions that fit a simple set of criteria. With selector tool technology, both the employers looking for employees, and consumers looking for positions, would be asked to fill out a detailed profile, which would be databased. Then, when either party is doing a search, they would go through the attribute selection, importance of difference and trade-off exercises; and the system would list all possibilities, in priority order as to how well they fit the preference profile. This would be much more efficient than currently available systems.

Yet another application of the invention is selection of assisted living facilities. The parents of millions of "baby boomers" are now at a time in their lives when they are considering full-care retirement centers, rest and nursing homes. There is no good source for them to search for centers which fit their needs economically, geographically and in terms of life-style needs. A web-based retirement home selection application would be a perfect use of the selector tool set. The company managing such a tool will database information on every retirement center and rest-home in America (or the world) and allow the users to click on what is important to them and make trade-offs. The tool would then list all centers which fit the preference profile of the user, in priority order as to how well they fit that person's desires.

Yet another application for the present invention is home selection. The MLS system has an existing database and an individual or couple could click on attributes ranging from price, distance from work and schools, style, etc. and quickly match their desires with homes that are available in any region they wish to explore. The selector tool would allow the user to select attributes that are important to the user, go through the importance and trade-off screen, and present the user with a listing of homes and corresponding utility values indicating which home most closely fits the user's preferences.

Still another application of the invention is selection of educational institutions. Choosing a college or university is always a challenge. There are books (Fiske and others) which offer a great deal of data, but searching is clumsy and unsophisticated on currently available web sites. The present invention could easily database college and university information, have links to their web sites and offer a powerful prioritizing service. Perhaps an even bigger market could be community-based, helping parents and students choose magnet schools, charter schools and other primary schools. This will become an increasingly big market because there is increasing interest in developing competition for schools in order to drive-up quality of schools through increased competition. School systems would fund implementation of a web-based educational institution selector tool, with research revenue and marketing fees to schools creating a powerful revenue stream.

Yet another application of the invention is an on-line dating service. Most on-line dating services ask a series of questions and perform a simple matching process. A selector tool would allow a wide variety of attributes to be selected by a user, importance of differences calculated between extremes and then trade-offs made. This could be the only on-line dating service which would show the user, in priority order, those people who fit them best. Such a tool could even include prototypical pictures of faces, bodies, etc for selection and trade-off.

Still another user for the present invention is to assist companies in making complex business decisions. Most major companies spend months and months, and thousands of dollars on staff and consultants to choose vendors for major hardware, software, builders, etc. The selector tool could be used to streamline evaluation of vendors in extremely complex situations.

As an example, a hospital is attempting to determine which new electronic medical records system should be purchased and implemented. Usually, a task force is established and an expensive consulting firm is retained. It takes months to specify the requirements, identify prospective vendors, interview vendors, write an RFP, evaluate the RFPs and make a selection. The tool would first be used with all key staff people within the hospital (finance, medical records, nurses, physicians, etc.), who would go through the selector tool in a mock selection. The selector tool will have in the system, every possible attribute to be considered in selection of a medical records system. All key people would take, for instance, 1 hour to thoughtfully go through the site, click on attributes, rate importance of difference and make trade-offs. The tool would then produce a report showing precisely what attributes are most important and even segmenting the results by divisions. This would greatly speed up the process of developing the RFP. After vendors respond to the RFP, real data from each vendor would be put into the selector tool and the same hospital staffers would go through the tool, this time to make a real selection. The selector tool would then show results for each users and aggregate the data from all users to show which system fits the needs of the hospital best. It would show every vendor, in priority order of fit, and then allow senior staff to configure and refine requirements and show which vendor is a best fit. This tool set could reduce decision time in half and increase consensus among staff member.

Other Business-to-Business Concepts

There are almost unlimited uses for the selector tool set for complex decision making in business. Below are just a few more concepts:

1. Helping employees with relocation
2. Choosing hospitals
3. Choosing physicians
4. Choosing consulting firms
5. Choosing architectural firms
6. Choosing commercial real estate
7. Choosing ISPs or ASPs
8. Choosing resorts
9. Choosing any benefits

Thus, as is apparent from the examples above, the selector tool according to the present invention greatly facilitates complex decision making in a variety of fields.

Scalability, Adaptability, and Efficiency

Another aspect of the present invention is its scalability and ability to adapt to different applications. FIG. 9 shown below is a block diagram of a preferred embodiment of a selector tool server according to the present invention. In FIG. 9, selector tool server 36A includes presentation layer **900**, business layer **902**, and data layer **904**. Presentation layer **900** contains functions for presenting text and graphics to the user. Business layer **902** includes components, which are separately compiled from application layer code, and extract and store data in data layer **904**. Data Layer **904** stores data, such as attributes and available products or services, for a given application.

Presentation layer **900** may be written in any suitable presentation layer language, such as Microsoft Active Server Pages (ASP). Microsoft ASP is a script-based language that uses HTML for presentation. The Present invention is not limited to using Microsoft ASP. Other languages that could be used include JSP—a Java-based language, or Cold Fusion. Presentation layer **900** is written to handle any client and any complex decision. Components that change from one application to another are data driven. For example, in order to modify the selector tool for use with a new application, it is not necessary to modify the code that creates the various graphical user interfaces. It is only necessary to modify data that changes from one application to the next, which is stored in data layer **904**. For example to change from a health plan selector tool to a 401(k) selector tool, it is only necessary to change the attributes and the available plans in data layer **904**.

According to another aspect of the invention, selector tool server **36A** runs session-less. For example, when user when the user changes from one screen to the next, server identification of the user was previously accomplished using a session-level variable. Currently, a temporary cookie is created in the user's browser when a user first accesses the tool. The cookie holds a unique identifier to identify the user as the user changes from page to page.

According to yet another aspect of the invention, no database calls done through presentation layer **900**. All calls to the database are done via components, which are part of business layer **902**. Using components to access data greatly reduces overhead. As discussed above, a component is a separately compiled piece of code with a dedicated function, such as "get product IDs available for this user." These components are called by presentation layer **900** when the function is needed. For example, for the attribute selection page, an attribute selection component extracts the attributes that are available for a specific user. Another component present in business layer **902** is the conjoint analysis engine, which is described in detail above.

According to another aspect of the invention, content, such as instructions for using the tool and other material displayable by the selector server, is preferably located in a content database in data layer **904**. Storing the content in a content database allows content to be quickly altered and customized for new clients or changing client needs. Data layer **904** may be implemented using any suitable database language, such as Microsoft Structured Query Language (SQL) Server Version 7.0.

According to yet another aspect of the invention, the underlying data schema has been created to be flexible and "generic." As used herein, the phrase "data schema" refers to the data structures and databases in data layer **904** used to generate the attributes and questions presented to the user. Rather than product or company-specific data structures, data structures are now generic. For example, rather that having a data structure created around the specific functional needs and specificities of a single client/product (i.e., health plans available to XYZ Corporation), a generic data schema has been developed that accommodates the data needs of the application regardless of the client and or product category. As a result providing generic data structures, the selector tool can be used for multiple clients and among multiple product categories, with little if any changes to the data schema.

According to another aspect of the invention, all frequently used query objects are indexed. A query object is a field in a database in data layer **904** that the selector tool might access. Indexing allows faster access to frequently accessed data.

As a result of the features listed above and factors relating to efficient coding, the selector tool server is capable of meeting the needs of multiple clients simultaneously. Current load testing demonstrates ability to handle 2,000 or more concurrent users.

It will be understood that various details of the invention may be changed without departing from the scope of the invention. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation—the invention being defined by the claims.

What is claimed is:

1. A method for facilitating user selection among complex alternatives using conjoint analysis, the method comprising:

using a software-implemented selector tool:

(a) obtaining demographic information from a user;

(b) executing a criteria matching heuristic, wherein the criteria matching heuristic automatically selects products or services and attributes associated with the products or services that are available to the user by accessing a first table that associates the demographic information with criteria identifiers and accessing a second table that matches criteria identifiers with products or services;

(c) presenting the user with a list of the attributes associated with the products or services available to the user and requiring the user to select among the attributes those which are of importance to the user;

(d) presenting the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting for each of the attributes selected by the user;

(e) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(f) responsive to receiving the first and second values from the user, applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each of the second values as a mathematical difference between the relative importance of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes;

(g) in response to completion of the automated conjoint analysis, rating the products or services available to the user based on the final importance values for the attributes linked to the corresponding stored product or service alternatives; and

(h) in response to completion of the calculating and the rating, automatically providing feedback from the conjoint analysis to the user, wherein providing feedback includes presenting the user with data indicating the relative utility to the user of each of the products or services.

2. The method of claim 1 wherein presenting the user with a list of attributes includes providing a graphical user interface for displaying the attributes to the user and receiving user selections of the attributes.

3. The method of claim 2 wherein providing a graphical user interface includes providing a graphical user interface using an application resident on a computer local to the user.

4. The method of claim 2 wherein providing a graphical user interface includes providing a graphical user interface using a server application resident on a computer remote from the user.

5. The method of claim 1 wherein requiring the user to input or select first values includes presenting discrete choices indicative of the relative importance of each attribute and requiring the user to select one of the discrete choices.

6. The method of claim 5 wherein presenting discrete choices to the user includes providing a graphical user interface having an importance of difference selection portion for allowing the user to select one of the discrete choices.

7. The method of claim 6 wherein providing a graphical user interface includes providing a graphical user interface using a program resident on a computer local to the user.

8. The method of claim 6 wherein providing a graphical user interface includes providing a graphical user interface using a server resident on a computer remote from the user.

9. The method of claim 1 wherein presenting the user with a second series of choices includes presenting the user with discrete choices indicating the relative importance of the difference between each pairing of attributes.

10. The method of claim 9 wherein presenting discrete choices to the user includes providing a graphical user interface having a difference in importance selection portion for allowing the user to select one of the discrete choices.

11. The method of claim 10 wherein providing a graphical user interface includes providing a graphical user interface using a program resident on a computer local to the user.

12. The method of claim 10 wherein providing a graphical user interface includes providing a graphical user interface using a server resident on a computer remote from the user.

13. The method of claim 1 wherein rating the products or services available to the user includes adding the final importance values of attributes associated with each product or services and presenting the sums to the user.

14. The method of claim 13 wherein presenting the sums to the user includes presenting the sums to the user using a graphical user interface.

15. The method of claim 14 wherein presenting the sums to the user using a graphical user interface includes presenting the sums to the user using a graphical user interface generated by an application program resident on a computer local to the user.

16. The method of claim 14 wherein presenting the sums to the user using a graphical user interface includes generating the graphical user interface using a server resident on a computer remote from the user.

17. The method of claim 1 wherein obtaining demographic information from the user includes presenting the user with a series of computer screens having questions relating to the user.

18. The method of claim 1 wherein the products or services include employer-sponsored health plans.

19. The computer program product of claim 1 wherein obtaining demographic information from the user includes presenting the user with a series of computer screens having questions relating to the user.

20. The method of claim 1 wherein the attributes and the product or services alternatives are stored in a database.

21. The method of claim 20 wherein the attributes and product or service alternatives comprise non-company-specific attributes and product or service alternatives.

22. A method for facilitating user selection among complex alternatives using conjoint analysis, the method comprising:

using a software-implemented selector tool:

(a) presenting a user with a list of attributes relating to a product or service and requiring the user to select among the attributes those which are of importance to the user;

(b) presenting the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting for each of the selected attributes;

(c) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(d) applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each of the second values as a mathematical difference between the relative importance of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes, wherein calculating a final importance value includes:

(a) generating a Y vector including the first values and the mathematical differences;

(b) generating an X matrix having a first portion indicating the ordering of the first values in the Y vector and second portion indicating the pairing of attributes corresponding to each mathematical difference in the Y vector;

(c) computing regression coefficients based on the Y vector and X matrix; and

(d) interpreting the regression coefficients as importance scores for attributes corresponding to the regression coefficients;

(e) in response to completion of the automated conjoint analysis, rating the products or services based on the final importance values; and

(f) presenting the user with data indicating the relative utility to the user of each of the products or services.

23. The method of claim 22 wherein calculating a final importance value for each of the attributes includes taking each regression coefficient as the value of the worst setting of an attribute and five times the final computed importance value as the best setting of the attribute and using linear interpolation to determine the final importance value.

24. The method of claim 23 wherein the final importance values for each of the attributes are unitless quantities.

25. A method for facilitating user selection among complex alternatives using conjoint analysis, the method comprising:

using a software-implemented selector tool:

(a) presenting a user with a list of attributes relating to predetermined products or services and requiring the user to select among the attributes those which are of importance to the user;

(b) presenting the user with a first series of choices requiring the user to input or select first values indi-

cating the relative importance of a best setting and a worst setting for each of the selected attributes;

(c) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(d) applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each of the second values as a mathematical difference between the relative importance of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes;

(e) responsive to completion of the conjoint analysis, rating the products or services based on the final importance values and stored links between the attributes and the products or services; and

(f) responsive to completion of the rating, providing feedback from the conjoint analysis to the user, wherein providing feedback includes presenting the user with data indicating the relative utility to the user of each of the products or services, wherein the predetermined products or services include mutual funds.

26. A method for facilitating user selection among complex alternatives using conjoint analysis, the method comprising:

using a software-implemented selector tool:

(a) presenting a user with a list of attributes relating to predetermined products or services and requiring the user to select among the attributes those which are of importance to the user;

(b) presenting the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting for each of the selected attributes;

(c) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(d) applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each second values as a mathematical difference between the relative importance of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes;

(e) responsive to completion of the conjoint analysis, rating the products or services available to the user based on the final importance values and stored links between the attributes and the products or services; and

(f) responsive to completion of the rating, providing feedback from the conjoint analysis to the user, wherein providing feedback includes presenting the user with data indicating the relative utility to the user of each of the products or services, wherein the predetermined products or services include consumer goods other than health plans.

27. A method for facilitating user selection among complex alternatives using conjoint analysis, the method comprising:

using a software-implemented selector tool:

(a) presenting a user with a list of attributes relating to predetermined products or services and requiring the user to select among the attributes those which are of importance to the user;

(b) presenting the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting for each of the selected attributes;

(c) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(d) applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each of the second values as a mathematical difference between the relative importance of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes;

(e) responsive to completion of the conjoint analysis, rating the products or services available to the user based on the final importance values and stored links between the attributes and the products or services; and

(f) responsive to completion of the rating, providing feedback from the conjoint analysis to the user, wherein providing feedback includes presenting the user, with data indicating the relative utility to the user of each of the products or services, wherein the products or services include medical services.

28. A method for facilitating user selection among complex alternatives using conjoint analysis, the method comprising:

using a software-implemented selector tool:

(a) presenting a user with a list of attributes relating to predetermined products or services and requiring the user to select among the attributes those which are of importance to the user;

(b) presenting the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting for each of the selected attributes;

(c) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(d) applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each of the second values as a mathematical difference between the relative importance

of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes;

(e) responsive to completion of the conjoint analysis, rating the products or services available to the user based on the final importance values and stored links between the attributes and the products or services; and

(f) responsive to completion of the rating, providing feedback from the conjoint analysis to the user, wherein providing feedback includes presenting the user with data indicating the relative utility to the user of each of the products or services, wherein the products or services include energy providers.

29. A method for facilitating user selection among complex alternatives using conjoint analysis, the method comprising:

using a software-implemented selector tool:

(a) presenting a user with a list of attributes relating to predetermined products or services and requiring the user to select among the attributes those which are of importance to the user;

(b) presenting the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting for each of the selected attributes;

(c) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(d) applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each second values as a mathematical difference between the relative importance of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes;

(e) responsive to completion of the conjoint analysis, rating the products or services available to the user based on the final importance values and stored links between the attributes and the products or services; and

(f) responsive to completion of the rating, providing feedback from the conjoint analysis to the user, wherein providing feedback includes presenting the user with data indicating the relative utility to the user of each of the products or services, wherein the products or services include assisted living facilities.

30. A method for facilitating user selection among complex alternatives using conjoint analysis, the method comprising:

using a software-implemented selector tool:

(a) presenting a user with a list of attributes relating to predetermined products or services and requiring the user to select among the attributes those which are of importance to the user;

(b) presenting the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting for each of the selected attributes;

(c) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(d) applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each of the second values as a mathematical difference between the relative importance of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes;

(e) responsive to completion of the conjoint analysis, rating the products or services available to the user based on the final importance values and stored links between the attributes and the products or services; and

(f) responsive to completion of the rating, providing feedback from the conjoint analysis to the user, wherein providing feedback includes presenting the user with data indicating the relative utility to the user of each of the products or services, wherein the products or services include educational institutions.

31. A method for facilitating user selection among complex alternatives using conjoint analysis, the method comprising:

using a software-implemented selector tool:

(a) presenting a user with a list of attributes relating to predetermined products or services and requiring the user to select among the attributes those which are of importance to the user;

(b) presenting the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting for each of the selected attributes;

(c) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(d) applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each of the second values as a mathematical difference between the relative importance of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes;

(e) responsive to completion of the conjoint analysis, rating the products or services available to the user based on the final importance values and stored links between the attributes and the products or services; and

(f) responsive to completion of the rating, providing feedback from the conjoint analysis to the user, wherein providing feedback includes presenting the user with data indicating the relative utility to the user of each of the products or services, wherein the products or services include potential employees or employers.

32. A method for facilitating user selection among complex alternatives using conjoint analysis, the method comprising:

using a software-implemented selector tool:

(a) presenting a user with a list of attributes relating to predetermined products or services and requiring the user to select among the attributes those which are of importance to the user;

(b) presenting the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting for each of the selected attributes;

(c) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(d) applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each of the second values as a mathematical difference between the relative importance of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes;

(e) responsive to completion of the conjoint analysis, rating the products or services available to the user based on the final importance values and stored links between the attributes and the products or services; and

(f) responsive to completion of the rating, providing feedback from the conjoint analysis to the user, wherein providing feedback includes presenting the user with data indicating the relative utility to the user of each of the products or services, wherein the products or services include potential companions available through an on-line dating service.

33. A computer program product comprising a computer executable instructions embodied in a computer readable medium for performing steps comprising:

using a software-implemented selector tool:

(a) obtaining demographic information from a user;

(b) executing a criteria matching heuristic, wherein the criteria matching heuristic automatically selects products or services and attributes associated with the products or services that are available to the user by accessing a first table that associates the demographic information with criteria identifiers and accessing a second table that matches criteria identifiers with products or services;

(c) presenting a user with a list of the attributes associated with the products or services available to the user and requiring the user to select among the attributes those which are of importance to the user;

(d) presenting the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting for each of the selected attributes selected by the user;

(e) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(f) responsive to receiving the first and second values from the user, applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each of the second values as a mathematical difference between the relative importance of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes;

(g) in response to completion of the automated conjoint analysis, rating the products or services available to the user based on the final importance values for the attributes linked to the corresponding stored product or service alternatives; and

(h) in response to the completion of the calculating and the rating, automatically providing feedback from the conjoint analysis to the user, wherein providing feedback includes presenting the user with data indicating the relative utility to the user of each of the products or services.

34. The computer program product of claim 33 wherein presenting the user with a list of attributes includes providing a graphical user interface for displaying the attributes to the user and receiving user selections of the attributes.

35. The computer program product of claim 34 wherein providing a graphical user interface includes providing a graphical user interface using an application resident on a computer local to the user.

36. The computer program product of claim 34 wherein providing a graphical user interface includes providing a graphical user interface using a server application resident on a computer remote from the user.

37. The computer program product of claim 33 wherein requiring the user to input or select first values includes presenting discrete choices indicative of the relative importance of each attribute and requiring the user to select one of the discrete choices.

38. The computer program product of claim 37 wherein presenting discrete choices to the user includes providing a graphical user interface having an importance of difference selection portion for allowing the user to select one of the discrete choices.

39. The computer program product of claim 38 wherein providing a graphical user interface includes providing a graphical user interface using a program resident on a computer local to the user.

40. The computer program product of claim 39 wherein providing a graphical user interface includes providing a graphical user interface using a server resident on a computer remote from the user.

41. The computer program product of claim 33 wherein presenting the user with a second series of choices includes presenting the user with discrete choices indicating the relative importance of the difference between each pairing of attributes.

42. The computer program product of claim 41 wherein presenting discrete choices to the user includes providing a graphical user interface having a difference in importance selection portion for allowing the user to select one of the discrete choices.

43. The computer program product of claim 42 wherein providing a graphical user interface includes providing a

graphical user interface using a program resident on a computer local to the user.

44. The computer program product of claim 42 wherein providing a graphical user interface includes providing a graphical user interface using a program resident on a computer remote from the user.

45. The computer program product of claim 33 wherein rating the products or services available to the user includes adding the final importance values of attributes associated with each product or services and presenting the sums to the user.

46. The computer program product of claim 45 wherein presenting the sums to the user includes presenting the sums to the user using a graphical user interface.

47. The computer program product of claim 46 wherein presenting the sums to the user using a graphical user interface includes presenting the sums to the user using a graphical user interface generated by an application program resident on a computer local to the user.

48. The computer program product of claim 46 wherein presenting the sums to the user using a graphical user interface includes generating the graphical user interface using a server resident on a computer remote from the user.

49. The computer program product of claim 33 wherein the products or services include employer-sponsored health plans.

50. The computer program product of claim 33 wherein the products or services include mutual funds.

51. The computer program product of claim 33 wherein the products or services include consumer goods.

52. The computer program product of claim 33 wherein the products or services include medical services.

53. The computer program product of claim 33 wherein the products or services include energy providers.

54. The computer program product of claim 33 wherein the products or services include assisted living facilities.

55. The computer program product of claim 33 wherein the products or services include educational institutions.

56. The computer program product of claim 33 wherein the products or services include potential employees or employers.

57. The computer program product of claim 33 wherein the products or services include potential companions available through an on-line dating service.

58. The computer program product of claim 33 wherein the products or services include combinations of complementary products or services.

59. The computer program product of claim 58 wherein the products or services include employee benefits.

60. A method for facilitating user selection among complex alternatives using conjoint analysis, the method comprising:

using a software-implemented selector tool:

(a) presenting a user with a list of attributes relating to a product or service and requiring the user to select among the attributes those which are of importance to the user;

(b) presenting the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting for each of the selected attributes;

(c) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(d) applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each of the second values as a mathematical difference between the relative importance of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes, wherein calculating a final importance value includes:

(a) generating a Y vector including the first values and the mathematical differences;

(b) generating an X matrix having a first portion indicating the ordering of the first values in the Y vector and second portion indicating the pairing of attributes corresponding to each mathematical difference in the Y vector;

(c) computing regression coefficients based on the Y vector and X matrix; and

(d) interpreting the regression coefficients as importance scores for attributes corresponding to the regression coefficients; and

(e) rating the products or services based on the final importance values; and

(f) presenting the user with data indicating the relative utility to the user of each of the products or services.

61. The computer program product of claim 60 wherein calculating a final importance value for each of the attributes includes taking each regression coefficient as the value of the worst setting of an attribute and five times the final computed importance value as the best setting of the attribute and using linear interpolation to determine the final importance value.

62. The computer program product of claim 61 wherein the final importance values for each of the attributes are unitless quantities.

63. A system for facilitating user selection among complex alternatives using conjoint analysis, the system comprising:

(a) a user interface generator for executing a criteria matching heuristic, wherein the criteria matching heuristic automatically selects products or services available to the user by obtaining demographic information from the user, accessing a first table that associates the demographic information with criteria identifiers and accessing a second table that matches criteria identifiers with products or services, the user interface generator for presenting screens to a user on a computer display device that require the user to rate the relative importance of the selected attributes for the selected products or services and for receiving input from the user regarding the attributes; and

(b) a conjoint analysis engine operatively associated with the user interface generator for receiving the user input for applying automated conjoint analysis to the user input, wherein applying automated conjoint analysis includes automatically calculating final importance values for each of the selected attributes and wherein the conjoint analysis engine is further adapted to calculate, based on the final importance values and the stored links between the selected attributes and the products or services, a total utility value for each of the products or services, and wherein the conjoint analysis engine is further adapted to provide feedback to the user based on the conjoint analysis, wherein providing feedback

includes delivering the total utility values to the user interface generator, which displays to the user the products or services with ratings based on the total utility values.

64. The system of claim **63** wherein the user interface generator is adapted to present the user with a first series of screens requiring the user to rate the relative importance between high and low settings of individual attributes.

65. The system of claim **64** wherein the user interface generator is adapted to present the user with a second series of screens requiring the user to rate the relative importance of the difference between a high setting of a first attribute and a low setting of a second attribute versus a low setting of the first attribute and a high setting of the second attribute.

66. The system of claim **65** wherein the conjoint analysis engine is adapted to interpret user input with regard to the second series of screens as a mathematical difference between the relative importance of the high setting and the low setting for first attribute and the high setting and the low setting for the second attribute.

67. The system of claim **63** wherein the user interface generator and the conjoint analysis engine comprise application programs adapted for execution on a server computer remote from the user.

68. The system of claim **63** wherein the user interface generator and the conjoint analysis engine comprise application programs adapted for execution on a computer local to the user.

69. The system of claim **63** wherein the products or services include employer-sponsored health plans.

70. The system of claim **63** wherein the products or services include mutual funds.

71. The system of claim **63** wherein the products or services include consumer goods.

72. The system of claim **63** wherein the products or services include medical services.

73. The system of claim **63** wherein the products or services include energy providers.

74. The system of claim **63** wherein the products or services include assisted living facilities.

75. The system of claim **63** wherein the products or services include educational institutions.

76. The system of claim **63** wherein the products or services include potential employees or employers.

77. The system of claim **63** wherein the products or services include potential companions available through an on-line dating service.

78. The system of claim **63** comprising:

(a) a presentation layer containing generic functions for generating the screens to be presented to the user;

(b) a data layer containing data relating to the attributes and the products or services available to the user; and

(c) a business layer for facilitating communication between the data layer and the presentation layer.

79. The system of claim **78** wherein the conjoint analysis engine is implemented in the business layer.

80. The system of claim **63** wherein the complex alternatives include real estate.

81. The system of claim **80** wherein the user is a prospective real estate buyer.

82. The system of claim **63** wherein the products or services include combinations of complementary products or services.

83. The system of claim **82** wherein the products or services include employee benefits.

84. A method for facilitating user selection among complex alternatives using conjoint analysis, the method comprising:

using a software-implemented selector tool:

(a) presenting a user with a list of attributes relating to predetermined products or services and requiring the user to select among the attributes those which are of importance to the user;

(b) presenting the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting for each of the selected attributes;

(c) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(d) applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each of the second values as a mathematical difference between the relative importance of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes;

(e) responsive to completion of the conjoint analysis, rating the products or services based on the final importance values and stored links between the attributes and the products or services; and

(f) responsive to completion of the rating, presenting the user with data indicating the relative utility to the user of each of the products or services, wherein the products or services include combinations of complementary products or services.

85. The method of claim **84** wherein the products or services include employee benefits other than health plans.

86. A method for facilitating user selection among complex alternatives using conjoint analysis, the method comprising:

using a software-implemented selector tool:

(a) presenting a user with a list of attributes relating to predetermined products or services and requiring the user to select among the attributes those which are of importance to the user;

(b) presenting the user with a first series of choices requiring the user to input or select first values indicating the relative importance of a best setting and a worst setting for each of the selected attributes;

(c) presenting the user with a second series of choices requiring the user to input or select second values indicating the relative importance of the user's preference between first and second pairings of the selected attributes, wherein each pairing includes a best setting of one attribute and a worst setting of another attribute;

(d) applying automated conjoint analysis to the first and second values, wherein applying automated conjoint analysis includes:

(i) interpreting each of the second values as a mathematical difference between the relative importance of a best setting and a worst setting for one attribute and a best setting and a worst setting for another attribute; and

(ii) calculating, based on the first values and the mathematical differences, a final importance value for each of the attributes;

(e) responsive to completion of the conjoint analysis, rating the products or services available to the user based on the final importance values and stored links between the attributes and the products or services; and

(f) responsive to completion of the rating, providing feedback from the conjoint analysis to the user wherein providing feedback includes presenting the user with data indicating the relative utility to the user of each of the products or services, wherein the products or services include 401(k) plans.

* * * * *

Related Proceedings Appendix

(none)